Quadtree-based Real-time Point Generalisation for Web and Mobile Mapping

---

Dissertation

zur

Erlangung der naturwissenschaftlichen Doktorwürde
(Dr. sc. nat.)

vorgelegt der

Mathematisch-naturwissenschaftlichen Fakultät

der

Universität Zürich

von

Pia Sandra Bereuter

von Illnau-Effretikon ZH

Promotionskomitee

Prof. Dr. Robert Weibel (Vorsitz)
Prof. Dr. Dirk Burghardt
Dr. William A. Mackaness

Zürich, 2015

Summary

Map generalisation, the process of drawing spatial phenomena to scale on maps, is a key process in cartography. The requirements of map generalisation differ depending on the output media and the context of usage. In web and mobile applications, in order to allow dynamic map interaction, real-time performance is crucial. In addition, modularity to flexibly adapt the map content is a second main requirement of map generalisation in the web and mobile context. With the advent of web and mobile mapping and the possibility to mash-up any kind of spatial data sources, point data, as the simplest but most abstracted form of spatial data, have gained importance. Despite today's increased prevalence of point data, real-time solutions for point data so far are clearly under-represented in the literature on generalisation.

The focus of this work therefore lies on the analysis and development of automated and real-time algorithms for cartographic generalisation of point data, streamlined to the needs of web and mobile mapping, as well as their integration as an interactive process in map portrayal. The unique feature of this thesis is that all proposed techniques are based on the quadtree, a hierarchical spatial data structure that has several properties that are very interesting for cartographic generalisation, which however have been used very rarely so far for this purpose.

The main contribution of this thesis is the development of a comprehensive conceptual framework and a modular system for real-time point generalisation operators and algorithms based on the quadtree data structure. The quadtree – more specifically, the point region (PR) quadtree – is exploited as a hierarchical spatial index and auxiliary data structure that enables both real-time performance and modularity and informs the generalisation algorithms and generalisation constraints. The use of the quadtree allows bridging the dichotomy of existing solutions in real-time map generalisation, which so far had been divided between the flexibility (but poor cartographic quality) of fast map generalisation algorithms and the speed (but inflexibility) of solutions relying on hierarchical data structures.

This work investigates several research questions, including: What characteristics are essential for an efficient and flexible real-time point generalisation algorithm to meet the requirements of web and mobile mapping? How can different types of point generalisation operators be integrated into a modular generalisation workflow? How can interaction in map generalisation be extended to enable dynamic online map information exploration? And finally, what are the strengths and weaknesses of quadtree-based algorithms for real-time point generalisation and how do they perform?

The thesis first identifies and categorises methods for real-time generalisation and develops a comprehensive, conceptual methodology for real-time point generalisation, as well as a problem-oriented, modular workflow for point generalisation. Subsequently, two main principles to approach point generalisation are introduced: on the one hand the so-called object-directed operators, which manipulate map objects directly; and on the other hand the space-directed operators, which manipulate and deform the map space in order to perform generalisation operations. The thesis proposes solutions for both object- and space-directed point generalisation operators.

For the main object-directed generalisation operators of point data — selection, simplification, aggregation, and displacement — a family of quadtree-based generalisation algorithms is proposed. For the case of space-directed generalisation two solutions that realize the concept of the 'malleable space' are presented: a density-equalising cartogram algorithm and Laplacian smoothing. For both algorithms, the quadtree informs the spatial deformation based on a scale-dependent density estimation.

This work furthermore contributes towards a more interactive use of map generalisation by introducing so-called 'content zooming' as a concept for visual exploration of content in web and mobile mapping. In combination with the quadtree-based point generalisation system, this solution to real-time generalisation not only illustrates the application of automated generalisation of point data, but also contributes to pushing interactive map generalisation towards a tool for visual map exploration.

Finally, this thesis presents numerous experiments and results with the proposed algorithms, and demonstrates how the various algorithms can be put together into complete workflows.

Zusammenfassung

Die Generalisierung von Karten ist ein Kernprozess in der Kartographie und beschreibt wie räumliche Phänomene abstrahiert und lesbar auf Karten dargestellt werden. Die Anforderungen an die Generalisierung sind dabei abhängig vom gewählten Darstellungsmedium und dem Kontext, in dem die Karten angewandt werden. Für webbasierte und mobile Anwendungen gelten hohe Anforderungen an eine Generalisierung, die in Echtzeit zu erfolgen hat. Für Nutzerinnen und Nutzer von online und mobilen Anwendungen ist zudem wichtig, dass der Karteninhalt dabei flexibel angepasst werden kann (Modularität). Durch das Aufkommen neuer Kartentypen, die es erlauben, Inhalte dynamisch aus mehreren Datenquellen zu sogenannten „mash-up“ Karten zusammenzustellen, haben Punktdaten an Wichtigkeit gewonnen. Punktdaten stellen die grundlegendste, aber zugleich auch am stärksten abstrahierte Form räumlicher Daten dar. Obwohl Punktdaten zunehmend auf Karten dargestellt werden, sind Lösungen zu deren Echtzeitgeneraliserung bis heute noch wenig erforscht.

Das Hauptaugenmerk der vorliegenden Arbeit ist deswegen die Analyse und Entwicklung von automatisierten Algorithmen, die in Echtzeit Punktdaten kartographisch generalisieren. Auf die spezifischen Bedürfnisse für web-basierte und mobile Karten und deren Integration in dynamische Darstellung wird dabei besonderen Wert gelegt. An dieser Arbeit ist speziell hervorzuheben, dass alle darin vorgeschlagenen Verfahren auf dem Quadtree beruhen, einer hierarchischen, räumlichen Datenstruktur. Obwohl der Quadtree für die kartographische Generalisierung interessante Eigenschaften aufweist, wurde diese Datenstruktur allerdings erst wenig für die Generalisierung verwendet.

Der Hauptbeitrag dieser Arbeit ist sowohl die Entwicklung einer umfassenden Methodologie, als auch eines auf dem Quadtree basierenden, modularen Systems von Operatoren und Algorithmen für die Echtzeit-Generalisierung von Punktdaten. Der Quadtree, besser gesagt der Point Region (PR) Quadtree, wird als hierarchischer, räumlicher Index und Hilfsdatenstruktur verwendet. Der PR Quadtree ermöglicht sowohl Echtzeitverhalten als auch Modularität der Algorithmen. Zusätzlich liefert er unterstützende Informationen, welche die Generalisierungsalgorithmen und -regeln unterstützen. Die Verwendung des Quadtrees erlaubt, die Dichotomie bisher aus der Literatur bekannten Ansätze zur Generalisierung in Echtzeit zu überwinden. Bisherige Ansätze zur Echtzeitgeneralisierung können in zwei Hauptansätze unterteilt werden. Dies sind einfache Generalisierungsalgorithmen mit hoher Flexibilität in der Prozessgestaltung, aber limitiert in der kartographischen Qualität und Lösungen basierend auf im Voraus berechneten, hierarchischen Datenstrukturen, von hoher kartographischer Qualität, die aber in der Flexibilität eingeschränkt sind.

Folgende Forschungsfragen waren für die Arbeit leitgebend: Welches sind notwendige Eigenschaften für eine effiziente und flexible Echtzeitgeneralisierung von Punktdaten,

welche die Anforderungen von Web- und Mobilkarten erfüllt? Wie können verschiedene Arten von Operatoren zur Punktgeneralisierung in einen modularen Arbeitsablauf integriert werden? Wie kann die Interaktion mit der Kartengeneralisierung erweitert werden, um eine dynamische Erkundung von Information in Webkarten zu ermöglichen? Und schliesslich, welches sind Stärken und Schwächen von auf dem Quadtree basierenden Algorithmen zur Echtzeitgeneralisierung, und wie verhalten sie sich?

Als Erstes erfolgt die Auswahl und Kategorisierung von Methoden zur Echtzeitgeneralisierung, die Entwicklung einer umfassenden Methodologie, als auch eines problemorientierten Arbeitprozesses der Generalisierung. Die Methodologie unterscheidet zwischen zwei Hauptprinzipien in der Punktgeneralisierung: Einerseits die objekt-gerichteten Generalisierungsoperatoren, welche die Kartenobjekte direkt manipulieren und andererseits die raum-gerichtete Operatoren, die den Kartenraum deformieren, um Generalisierungsoperationen durchzuführen. Der in der Arbeit entwickelte Prototyp implementiert sowohl Lösungen für objekt- und für raumgerichtete Generalisierungsoperatoren.

Die wichtigsten objekt-gerichteten Generalisierungsoperatoren von Punktdaten sind Selektion, Vereinfachung, Aggregation und Verdrängung. Für diese Hauptoperatoren der Generalisierung schlägt die Arbeit quadtree-basierte Generalisierungsalgorithmen vor. Für den Fall der raumgerichteten Generalisierung sind zwei Ansätze umgesetzt worden, welche das Konzept des dehnbaren Raumes ('malleable space') verwenden. Erstens dichtebasierte Kartogramme und zweitens Laplace Glättung. Der Quadtree ist bei beiden Algorithmen ein masstabsabhängiger Dichteschätzer für die räumliche Deformation.

Acknowledgements

This work, carried out at the Department of Geography of the University of Zurich, would not have been possible without the great support of numerous people and institutions. I warmly thank all those people for their help.

I am most grateful to:

Robert Weibel, as my main supervisor, for giving me the opportunity to carry out this research in the GIS group. Especially for his support, patience, readings, suggestion and good, in-depth discussions.

Dirk Burghardt, who acted as my second supervisor and initiated together with Robert Weibel the GenW2 project. He owes my thanks for his supervision and support, as well as for the productive research stay at Dresden University of Technology in September 2011.

Ramya Venkateswaran for being a good companion in the GenW2project and especially for her patience with all my programming questions.

William Mackaness (University of Edinburgh) for acting as external member of my PhD committee and Lars Harrie (Lund Unversity) for acting as external reviewer of this thesis.

Silvia Stofer of the National Data and Information Centre for Swiss Lichens and all the contributors for collecting and providing the data of the SwissLichens dataset and providing a perfect dataset for testing the generalisation algorithms.

My colleagues of the three GIScience units at the Department of Geography for providing such a great research culture and such stimulating discussions.

.. and most of all my family, friends and especially my partner for their patience, understanding and being a great support during the time of dissertation.

Pia Bereuter, Zürich, October 1, 2014

This research is funded by the Swiss National Science foundation (SNF) through the projects GenW2 and GenW2+ (grant no. 200020-120256, 200020-138109). Financial support is gratefully acknowledged.

Contents

List of Figures

List of Tables

Chapter 1.

Introduction

1.1. Motivation

A world without interactive maps on our computers or mobile phones is hard to imagine nowadays, if it ever was possible to imagine a world without any sort of map as a language to describe space. What we can imagine, though, is how we picture future maps and map interaction. Looking at recent development in geographic information science and cartography, the current efforts aim to take them a step further, towards maps that serve as a spatial, interactive, dynamic and adaptive spatial communication tool.

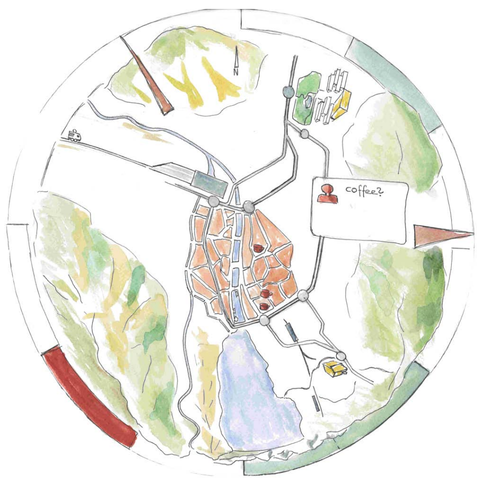


Figure 1.1.: Sketch of an adaptive mapping interface (Pia Bereuter)

With the rise of digital technologies over recent decades, map use and map production has changed tremendously. Demands and new possibilities in map use and map making motivate and benefit from the continuous development in technology and science. Whereas twenty years ago one would mainly navigate with paper maps and would need sufficient skills in map reading, route planning and self location, nowadays one is more likely to use a mobile phone, to find the present position and get route instructions, enhanced with further information about the current location.

This change in technology had a strong influence on the closely linked fields of *cartography* and *geography*. *Geographic information systems (GIS)* as a tool in scientific research and *geographic information science (GIScience)* as a field, are the consequent response within the domain of geography to this fundamental shift in technology. The demarcation of geographic information science as a science, and its definition, has been debated ever since its first mention (see Mark, 2003; Goodchild, 2010, and many others). Geographic information science is a child of its time and caused a paradigm shift of how geographic information is acquired, processed and visualised. A similar change happened in cartography. The technological shift fundamentally changed how maps are created, how map data is modelled and generalised, how it is displayed and especially how users interact with and through maps.

Location based services Effects of miniaturisation in technology enable mobile computing devices such as mobile phones, tablets or wearable computers. The inclusion of embedded positioning systems, such as integrated GPS modules, link a device to its geographical location. In mobile GIS, location-based services (LBS) provide – based on the position of the device – personalised geographical data and information services, most often in the form of maps.

In addition to the concept of interactivity of desktop and web mapping, the concept of mobility in mobile cartography allows the adaptation of maps to the needs of the mobile user. Mobile mapping induces certain constraints, which map design has to account for, such as small screen size, exposure to environmental conditions, limited computing power and bandwidth, and potential network failure. For a more user-centric approach, maps can be designed to be more 'egocentric' by emphasising context adaptation and personalisation.

Web 2.0 The major advantage of the world wide web is that it directly connects content provider and consumer. It directly links writer and and reader, by means of a URL and a browser. While in the beginning of the world wide web the content was rather static, over time the websites became more interactive and collaborative, providing web-based applications and virtual communities to the users.

The terms *Web 2.0* and *mash-ups* describe this change and characterise the involvement of users as content producers and the creation of large online social networks. On a technological level, web services, different application programming interfaces (API), and client side scripting (Ajax, JavaScript frameworks, HTML5) enabled the design of dynamic, user-friendly and interactive web interfaces. Furthermore Web 2.0 with its participatory aspect promotes machine-readable data exchange formats, such as XML or JSON (and GeoJSON).

The possibility to link and integrate content from different sources and services led to hybrid applications, so-called *mash-ups*, providing a tailor-made experience to the user (Butler, 2006; Miller, 2006). Online mapping applications such as Google, Yahoo, Bing, Nokia and OpenStreetMap often provide a public API to access their web services. The first published map mash-up was a reverse engineered Google Map overlaid with a housing dataset of point data, by Paul Rademacher in 2005, before the Google Maps API was released later that year (Miller, 2006).

Tourguide mash-up A proof of concept in the form of a tourism application prototype for Paris exemplifies the potential of web and mobile mapping combined with data integration from various sources (Venkateswaran and Bereuter, 2009). The application integrates the Metro timetable, tourist attractions, and restaurants and their opening times, and links this information to the social network profiles of the user (Figure 1.2). The

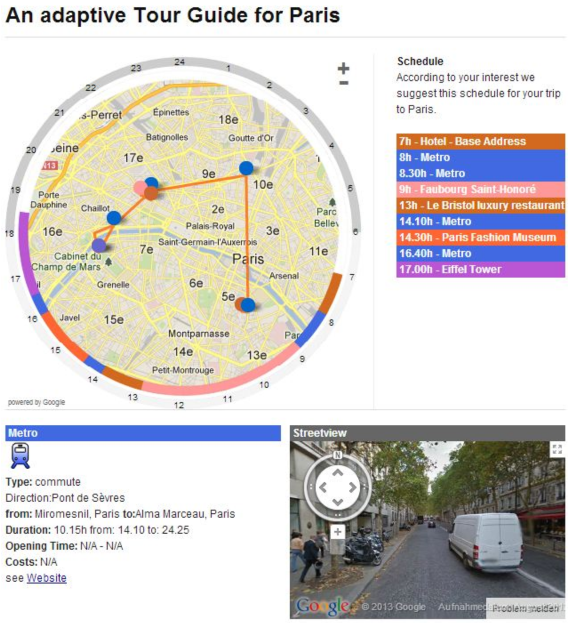


Figure 1.2.: Mash-up application prototype: adaptive tour guide for Paris

application suggests important places to see and creates a tour through the city of Paris, based on several factors and the various included data sources. The factors included were activities and interests derived from the user's Facebook profile, travel habits from the user's Dopplr profile, as well as travelling dates, base address, mode of transport, budget and purpose of travel.

The proposed tour is presented to the user through an interactive map. The representation integrates different data services. It features a tour map and a time schedule with a clock-like representation showing the timetable of the tourism events during a day. For each event it presents the details of the event and, if available, Google Street View data of the event location. Mash-up applications like this one not only allow the linking of different data sources, but consequently enable new forms of data visualisation and interaction.

Despite the rapid evolution of technology and new opportunities in spatial communication, the adaptation to context, given tasks and user, are not exploited completely so far. Hence, adaptation in spatial communication remains a major research topic in the area of mobile computing and LBS.

As a consequence of this evolution, the map user is presented with an increasing variety of map products. From paper maps to digital maps including GIS, from general-purpose digital maps, to adaptable Web 2.0 maps, such as mash-ups combined with volunteered content, and recently from adaptable maps to adaptive maps in the mobile context. The domain of map creation has ceased to be solely in the hands of experienced cartographers, and has been adopted by multimedia artists and different research fields, creating new forms of visualisation and portrayal – ultimately enriching the domain of cartography.

Map generalisation for web and mobile maps With each new output medium new opportunities and limitations emerge, which is especially true for map generalisation, a key process in cartography. In map generalisation spatial phenomena are drawn to scale on maps. Changes to map making and new technical opportunities alter the preliminaries of map generalisation and induce a need for generalisation functionality in graphical information delivery and portrayal. The requirements for map generalisation differ depending on the output media and the context of usage. Mackaness and Ruas (2007) state that irrespective of the media or the degree of interaction, the power of maps remains in their ability to abstract geographic space. Hence, different levels of abstraction reveal different patterns and properties, inherent to the geographic phenomena being represented. The authors further emphasise that the ability to abstract data is even more important in today's information society – in which the volume of data exceeds our insatiable appetite for more. Furthermore, in web and mobile applications, generalisation takes place in an interactive environment and has to be in sync with context adaptation and personalisation. In order to allow dynamic map interaction, real-time performance is key, especially given that the visualisation takes place on small screens and the processing power of the mobile mapping clients is limited.

Most of the data that is visualised on maps in Web 2.0 mash-ups comes in the form of point data. However, despite the increased prevalence of point data, real-time solutions for point data so far are under-represented in the literature on generalisation. Thus, the scope of this work lies in the analysis, development and integration of automated and real-time cartographic point data generalisation algorithms, streamlined to the needs of web and mobile mapping, and their integration as an interactive process in map portrayal – the automatic generation of interactive maps.

1.2. Objectives and research questions

This thesis is embedded in the context of the Swiss National Science Foundation (SNSF) project *Generalisation for Portrayal in Web and Wireless Mapping (GenW2)* with the main objective to contribute to the development of methods for automated generalisation in mobile and web-based map portrayal. GenW2 consists of two sub-projects addressed by two separate PhD theses. The first sub-project investigates ad-hoc integration of heterogeneous data sources for web and mobile GIS applications (Venkateswaran, 2015), while the second sub-project is represented by this thesis. This work extends from the results of several preceding PhD theses (Cecconi, 2003; Edwardes, 2007; Neun, 2007; Steiniger, 2007).

The main objective of this thesis is to develop dynamic real-time generalisation algorithms for point data and to integrate point generalisation with human computer interaction (HCI) techniques. It pursues the following research questions:

Real-time point generalisation algorithms: *What characteristics are essential for an efficient and flexible real-time point generalisation algorithm?*

The first research question is motivated by what is needed to enable real-time generalisation, such that it meets the requirements of web and mobile mapping. It is addressed in Chapters 6 and 3, respectively.

Space-directed point generalisation algorithms: *How can different types of point generalisation operators be integrated into a modular generalisation workflow?*
This work introduces two main principles in approaching point generalisation (see Chapter 3). On the one hand there are so-called object-directed operators (see Chapter 6), which manipulate map objects directly, and on the other hand space-directed operators, which manipulate the map space in order to perform generalisation operations. Chapter 7 presents solutions to space-directed point generalisation. How these operators can be integrated into a modular generalisation workflow is presented in Chapter 9.

Map interaction and generalisation: *How can interaction in map generalisation be extended to enable dynamic online map information exploration?*
This research question addresses the problem on how generalisation can assist the interactive exploration of map information. Chapter 4 pursues this question and presents a novel methodology for information exploration in web and mobile maps.

Generalisation evaluation: *What are the strengths and weaknesses of quadtree-based algorithms for real-time point generalisation and how do they perform?*
This thesis proposes the quadtree data structure as a possible index structure to facilitate real-time generalisation. The means to assess the cartographic quality of point generalisation are introduced in Chapter 5, while Chapter 8 presents the cartographic analysis of the quadtree-based generalisation approach.

1.3. Outline of the thesis

The content of this thesis is organised into eleven chapters, grouped into four thematic parts:

Research context and methodology Following this introductory chapter the first part of this thesis starts with the presentation of the theoretical background in Chapter 2, introducing the principles of map generalisation, with a focus on real-time map generalisation. The chapter also reviews the state of the art and identifies research gaps.
Chapter 3 introduces the research methodology for real-time point generalisation, as well as a classification of point generalisation operations that serves as a conceptual basis for this work.

Interaction techniques and methods of cartographic analysis Chapter 4 introduces a map interaction technique called content zooming, to be used for content exploration, as a technique of changing the degree of abstraction of map content independently of the map scale.
Chapter 5 presents a diagnostic toolbox that enables the quantitative measurement and evaluation of the strengths and weaknesses of point generalisation with regards to desirable cartographic properties. The methods of the toolbox provide the basis

for the analysis of the proposed real-time generalisation algorithms in the subsequent chapters, in particular chapter 8.

Real-time generalisation and its design This part contains the main substance of this thesis, proposing a variety of algorithms and workflows for point data generalisation. Chapter 6 introduces object-directed real-time generalisation algorithms for point data, while Chapter 7 does the same for their space-directed equivalents. The descriptions include the presentation of associated storage structures to facilitate efficient interactive zooming and panning. Chapter 9 then shows the integration of the proposed individual algorithms into overall workflows of the generalisation process (following Ch. 8).

Experiments and analysis A set of quantitative and qualitative cartographic experiments form the core of Chapter 8. It aims to thoroughly compare and analyse the different algorithms, and investigate their performance, scaling behaviour and storage options.

Chapter 10 provides a detailed discussion of the experimental results as well as the strengths and weaknesses of the proposed methodology and algorithms. The chapter concludes by answering the research questions of this thesis. In the final Chapter 11 the main contributions of this thesis as well as the insights gained are highlighted and an outlook on future work is offered. The design of the developed prototype is described in Appendix A. Appendix B provides descriptions of the data sets used in the experiments, while Appendix C details a questionnaire employed to solicit user feedback in evaluating the settings of a particular algorithm.

This thesis and its chapters are based on and extend from research published in peer-reviewed conference proceedings and scientific journals, as listed below. A complete list of publications is provided in the Appendix.

Methodology

Bereuter, P. and Weibel, R.(2010). Generalisation of point data for mobile devices: A problem-oriented approach. In S. Mustiere and W.A. Mackaness, editors, *13th Workshop on Progress in Generalisation and Multiple Representation*, *ICA Commission on Generalisation and Multiple Representation*, pages 1–8. Zurich, Switzerland.

Interaction techniques

Bereuter, P., Weibel, R., and Burghardt, D., (2013). Content zooming and information exploration for mobile maps. *International Journal of Geomatics and Spatial Analysis*, Revue internationale de géomatique, 23(3-4), 295–321.

Real-time generalisation

Bereuter, P. and Weibel, R. (2013). Real-time generalization of point data in mobile and web mapping using quadtrees. *Cartography and Geographic Information Science*, 40(4):271–281.

Experiments and analysis

Bereuter, P. and Weibel, R. (2011). A Diagnostic Toolbox for Assessing Point Data Generalisation Algorithms. In *Proceedings of the 25th International Cartographic Conference.*, Paris, France, 1–10.

Bereuter, P. and Weibel, R.(2012). Assessing Real-Time Generalisation of Point Data. In *16th Workshop on Progress in Generalisation and Multiple Representation map generalization.*, Dresden, Germany, 1–9.

Stanislawski, L. V, Buttenfield, B., Bereuter, P., and Brewer, C., 2014. Generalisation Operators. In: D. Burghardt, C. Duchêne, and W. Mackaness, eds. *Abstracting Geographic Information in a Data Rich World: Methodologies and Applications of Map Generalisation*, Springer International Publishing, 157–195.

Chapter 2.

Related Work

2.1. Map generalisation

Within the domain of cartography, map generalisation is a key element and fundamental process. In map generalisation spatial phenomena are drawn to scale on maps, designed with various purposes and thematics, such as, for instance, topographic, geological or road maps. In drawing spatial objects to scale, the available physical space is reduced. It follows that small objects may reach the limits of visual perceptibility and their representation has to be either visually enlarged or removed. Enlarged map objects, or overlying map features, as a consequence, compete for limited map space. This implies that the conveyed map information has to be carefully selected and abstracted. With the aim of maintaining the clarity and the aesthetic quality of a map, generalisation keeps the relevant information for a specific scale, reducing its complexity by suppressing the insignificant.

> Generalisation – the selection and simplified representation of detail appropriate to the scale and/or the purpose of the map (International Cartographic Association, 1973).

McMaster and Shea (1992) extend the above definition and include the map audience. The ICA definition of generalisation still holds nowadays, since the aim of map generalisation has not changed. The basic conditions, such as the technical opportunities, however did. Following the conceptual framework for automated map generalisation proposed by Brassel and Weibel (1988,page 102), what has also changed is the possibility to formulate 'controls' that govern the generalisation process. Controls include generalisation objectives and scale or communication rules, which are considered important ingredients for generalisation and allow for a more interactive and dynamic use of maps. Within the McMaster and Shea (1992) model, the controls are covered by the philosophical objectives – the why – inside the subcategory of application-specific elements.

Compared to traditional (paper) maps, the creation of digital maps and map interaction can be designed in a more flexible and modular fashion. The cartographic quality compared to traditional paper maps, is often more relaxed. Maps – especially web- and online maps – tend to be used more ephemerally and are therefore often created on demand, for a momentary use. Mobile and web 2.0 maps often integrate various heterogeneous data sources dynamically. The interaction and content adaptation of those maps is often designed for specific map purposes and the map is often integrated into a broader information flow and context. In such a highly interactive setting real-time behaviour becomes crucial, especially if the generalisation of the data is a requirement.

The rise and prevalence of location-based technologies such as GPS navigation devices, not only provides new opportunities for mapping application, but it also provides an increased ease in capturing location, and therefore simplifies data collection in the field. This has ultimately led to an increased availability of geo-spatial data and a predominance of point data, as point data are frequently the easily collected form of geo-spatial data.

Jones and Ware (2005) define map generalisation in the context of real-time generalisation, as a process of adapting the content and the LOD of a map to suit scale and purpose. The usage of mobile mapping services in a highly dynamic environment requires adaptation of content to different changing contexts. Furthermore, for devices with a small screen, such as mobile phones, Jones and Ware (2005) state that, "it is all the more desirable that the level of generalisation be adapted flexibly to meet the needs of an individual user" (Edwardes and Burghardt, 2003; Jakobsson, 2003). User interaction needs to be minimized and services adapted to become context sensitive and personalized (Reichenbacher, 2004; Duckham et al., 2006; Raper, 2007; Raubal and Panov, 2008).

Recent work also investigates the notion of geographic relevance within the context of location based services (Reichenbacher and De Sabbata, 2011; De Sabbata and Reichenbacher, 2012; Raper, 2007), assigning relevance to geographic features-based on its context beyond its mere location. In the context of map generalisation, this notion of geographic relevance, can be seen as a data enrichment process prior to the generalisation process. Other limitations that need to be considered by the dynamic use of various mobile services are transparency, privacy and obfuscation (Duckham et al., 2006). Despite the rapid evolution of techniques, capabilities available for spatial communication are not exploited completely, such as adaptation to context, given tasks and the dynamic user profile.

Map generalisation, key insights

- Generalisation, especially real-time generalisation, should aim to adapt map content and the LOD of a map to suit scale and purpose.
- Generalisation should adapt flexibly to the needs of an individual user.

2.2. Automated map generalisation

Research in automated generalisation seeks to leverage the labour intensive manual work of traditional cartographic generalisation. The aim of automated generalisation is to capture geographic data only once at the most detailed scale and subsequently derive all scales and map products. Conceptual map generalisation frameworks model the map generalisation process. These frameworks serve as a basis for understanding, in order to develop automatic generalisation solutions and to identify essential components and processes. A thorough overview on manual generalisation and early generalisation framework is given by McMaster and Shea (1992); Weibel and Dutton (1999) and Sarjakoski (2007).

Two of the first conceptual frameworks developed for automated generalisation are the conceptual framework of Brassel and Weibel (1988) and the one of McMaster and Shea (1992). The Brassel and Weibel model distinguishes between statistical generalisation (in later publications termed model generalisation (Weibel, 1995)) and cartographic generalisation, and describes a five step process consisting of: structure recognition, process recognition, process modelling, process execution and data display. The extension of the conceptual framework into a comprehensive digital generalisation model by McMaster and Shea (1992) subdivides the generalisation process into three critical questions necessary to successfully conduct map generalisation: *why to generalise* by adhering philosophical objectives, *when to generalise* by cartometric evaluation and *how to generalise* by the implementation of spatial attribute transformation.

Grünreich (1992), within the development of the ATKIS (Amtliches Topographisch-Kartographisches Informationssystem) in Germany, subdivides the digital map generalisation process into three different processes: object generalisation, model generalisation and cartographic generalisation. Object generalisation indicates the building of the primary high resolution geographical database as an abstract representation of the world. Model generalisation, also termed statistical generalisation (Brassel and Weibel, 1988), is a controlled reduction of details that emphasizes larger landscape features. Model generalisation can be seen as a pre-processing and filtering step for cartographic generalisation, without dealing with visualization issues. Cartographic generalisation then denotes the process, where a map is derived to scale for its portrayal, what is usually referred to as generalisation.

While generalisation algorithms solve one particular generalisation task of a specific generalisation operator (cf. Section 2.3), an overall generalisation process requires an orchestration of separate tasks, and procedural knowledge to be properly executed. Or as Weibel and Dutton (1999) state;

> If a generalisation system were-based solely on the data models and algorithms [...], much of the higher-level reasoning and decision strategies would be lacking, making it necessary to rely entirely on the user for the provision of this missing knowledge. (Weibel and Dutton, 1999)

Several approaches have been applied in literature to achieve an automated control of the generalisation process. Harrie and Weibel (2007) review existing methods and distinguish three modelling techniques: condition-action modelling, human interaction modelling and constraint-based modelling used to form a comprehensive generalisation process.

In condition-action modelling rules are formalised as condition-action pairs, which requires the explicit formulation of often implicit and intuitive cartographic knowledge. Human interaction modelling, however, incorporates the cartographer's knowledge into the generalisation process by sharing the cognitive workload between computer and human, whereas the generalisation system provides decision support, and the human operator his procedural knowledge. Both approaches, condition-action modelling and human interaction modelling, have their disadvantages: while the weakness of condition-action modelling lies in the formulation of cartographic knowledge (Compton and Jansen, 1990), large number of rules, lack of flexibility and no capability to evaluate cartographic results (Mackaness and Ruas, 2007), human interaction modelling systems are highly dependent on the human operator (Ruas, 2001).

Based on the shortcomings of earlier automated generalisation processes, Beard (1991) therefore proposed constraint-based modelling for automated generalisation, by adapting an approach from computer science to map generalisation. A later model by Ruas and Plazanet (1996) partly based on the Weibel and Brassel model introduces an iterative constraint-based approach, motivated by the difficulty in rule-based systems to acquire and formalize cartographic rules consistently. Constraints, in comparison to rules that state what has to be done within a process, stress what results have to be obtained Harrie and Weibel (2007). Constraints, in comparison to measures, not only characterize map objects, but also formulate map objectives and govern a particular generalisation task (Beard, 1991; Weibel, 1997).

Table 2.1.: Typology of constraints, and their property adapted from Harrie and Weibel (2007)

Constraint	Description	Single objects	Object groups
Representation constraints			
Position	Ensures the position of map objects, either absolute within the geodetic reference system or relative by maintaining the distance between map objects.		
Topology	Ensures the topological relationship between map objects.		
Shape	Maintains essential characteristics of single objects (e.g. area and angularity preservation).		
Structural	Preserves object patterns (e.g. road networks) by object alignment, mean distance and distribution of objects.		
Functional	Maintains map purpose e.g. correct representation of the road network of road maps		
Map readability constraints			
Legibility	Preserves map readability, with no spatial conflict, large enough and not too detailed features conform to graphic limits.		

Map constraints are led by the two main requirements a generalised map has to fulfil (Harrie and Weibel, 2007): representation constraints and map readability constraints. The final generalised map is then supposed to be a compromise between good readability and good representation. The introduction of constraint-based modelling led to the creation of new solutions to guide and implement the generalisation process and provide the possibility of self evaluation of a generalisation process. Harrie and Weibel (2007) distinguish between three constraint-based modelling approaches: agent modelling, combinatorial optimisation modelling and continuous optimisation modelling. Constraint-based approaches were furthermore proposed for a workflow-based generalisation system (Petzold et al., 2006).

Agent modelling addresses constraint violations with the help of multiple autonomous, "intelligent" agents which try to find an optimal state with low constraint violations (Ruas, 1998; Regnauld, 2001; Barrault et al., 2001). Combinatorial optimisation modelling applies methods such as simulated annealing (Ware et al., 2003) or genetic algorithms (Wilson et al., 2003) to find an optimal generalisation solution based on predefined map constraints. Continuous optimisation modelling applies measures, based on a set of constraints, to form an objective function where the minimal value is used as an optimal solution for the generalisation process, such as snakes (Burghardt and Meier, 1997; Burghardt, 2005; Bader, 2001), elastic beams (Bader, 2001) or solutions based on least square adjustment (Sester, 2000; Harrie and Sarjakoski, 2002).

For the domain of web- and mobile mapping, Harrie and Weibel (2007) state that to date, even with a distributed generalisation approach (Burghardt et al., 2005), the modelling of the generalisation approach lies within the responsibility of the client. They suggest that a specification of generalisation tasks at higher level, with the integration of constraints with new interfaces, may make a shift of the modelling responsibility to the server possible.

Automated map generalisation, key insights

- Generalisation process requires an orchestration of its single tasks and procedural and structural knowledge to be properly executed.
- Constraint-based modelling focuses on the generalisation objectives rather than on simple measures, however it highly depends on the definition of each constraint.
- Modelling approaches followed such as continuous and combinatorial optimisation or agent modelling, are computationally rather expensive. More efficient constraint-based approaches are required.

2.3. Generalisation operators

It is generally assumed that it is possible to break down the generalisation process into a chain of logical operations (Regnauld and McMaster, 2007). Generalisation operators define a particular generalisation process in a conceptual way; they are implemented by generalisation algorithms. For a given operator, several algorithms are usually possible (McMaster and Shea, 1992). Depending on the author, they were sometimes also referred to as *spatial & attribute transformations* (McMaster and Shea, 1992) or *generalisation processes* (Brassel and Weibel, 1988). The relationship between generalisation and algorithm is hierarchical (Weibel, 1997). Generalisation operators define the transformation to achieve, whilst the algorithm implements the requested transformation. This implies that the generalisation operator is independent of a particular data model (Weibel and Dutton, 1999). The algorithms, however, are specific to the data model, purpose and feature type; they often also depend on the geographic phenomena the map features depict. During the generalisation process, for each operator, scale, and data type, the optimal algorithm is chosen. For most operators several algorithms have been developed, each with its specific characteristics. A good example are algorithms for line simplification. It is important to note that there is a fundamental difference between algorithms designed for object/vector based models and those for raster based data models (Regnauld and McMaster, 2007). As most operators depend on the topology between map objects, raster-based algorithms are straightforward to implement, whereas vector-based algorithms need to model the spatial relationship of map objects.

Various typologies for generalisation operators have emerged since the early typologies of manual map production (Brassel and Weibel, 1988; McMaster and Shea, 1992; Weibel and Dutton, 1999; Regnauld and McMaster, 2007; Roth et al., 2007). However there is no general agreement on their terminology and description. The typology of generalisation operators byMcMaster and Shea (1992) with its focus on line generalisation distinguishes between attribute and spatial transformation, and among the spatial transformation between the data models. Their typology within spatial transformation differentiates between location-based raster generalisation and object-based vector generalisation.

Weibel and Dutton (1999) in contrast, highlight the relationship between generalisation operator and data (type) – explicitly not intended to be exhaustive – and listed the following operators valid for all data types: *eliminate/select, simplify, smooth, aggregate/ amalgamate/ merge, collapse, displace, enhance/ exaggerate*.

The AGENT project (Barrault et al., 2001; AGENT, 1999) extends this typology further and distinguishes between traditional operators and their derived digital operators. Regnauld and McMaster (2007) provide an overview of the different frameworks and a consolidated list, while Roth et al. (2007; 2011) explicitly include symbol changes as a generalisation and differentiate between, content, geometry and symbology operators.

All of these generalisation operators described have in common the feature that they change the map objects themselves (points, lines and polygons) and do not deform the underlying map space (e.g. fisheye distortion) to solve cartographic conflicts (Chapter 3).

Continuous optimisation methods used in cartographic object displacement, such as elastic beams (Bader, 2001; Bader et al., 2005) or least squares adjustment (Sester, 2000; Harrie and Sarjakoski, 2002) implement the displacement process as a continuous transformation, in order to move overlapping foreground objects away from each other. Thus, these methods could in principle also be used to continuously deform the map space, implementing the concept of the malleable space. However, in the literature so far their focus lies on manipulating the foreground objects in order to satisfy the constraints of the displacement operator, rather than the map foreground *and* background.

As for the generalisation of point maps Edwardes et al. (2005) identified selection, simplification, aggregation, typification and displacement, as the relevant and most suitable operators for point generalisation. Selection, simplification, typification and aggregation reduce the number of features represented on the map, while displacement moves the map features away from each other to remove overlaps and congestion. Among the point reduction operators, selection and simplification choose a subset of the original points, while aggregation typically generates new point positions (placeholders). The difference between selection and simplification is the same as in the classification by McMaster and Shea (1992): selection is based on attributes, while simplification is based on geometric criteria. Each of the above generalisation operators has different requirements.

However, for all operators the detection of spatial conflicts is crucial. For selection and aggregation, hierarchical ordering plays a key role, while displacement necessitates region and neighbourhood queries. A hierarchical data structure like the quadtree has both: hierarchical spatial order and the facility to speed up region and neighbourhood queries.

Generalisation operators, key insights

- Generalisation operators define a particular generalisation process in a conceptual way and are implemented by generalisation algorithms.
- Generalisation operators operate on map objects and do not manipulate the underlying map space.
- Generalisation operators should be independent of the applied data model.
- Relevant generalisation operators for point data identified are selection, simplification, aggregation, typification and displacement.

2.4. Real-time or on-the-fly Generalisation

2.4.1. General Overview

In the described context of web- and mobile mapping, with the increased availability of geo-spatial data, real-time or on-the-fly generalisation becomes a must. Van Oosterom (1995) describes on-the-fly generalisation as the automatic, on-the-fly derivation of a temporary display from a single detailed dataset to avoid redundancies. An even shorter definition is given by Weibel and Burghardt (2008) stating that on-the-fly generalisation denotes automatic generalisation in real time. *Real-time* and *on-the-fly* map generalisation are used as synonyms in the literature of map generalisation. This thesis employs the term *real-time generalisation* for being the more self-explanatory term.

Real-time generalisation for mobile mapping is typically achieved by relying either on fast generalisation algorithms or on pre-computation and hierarchical data structures (Weibel and Burghardt, 2008; van Oosterom, 2005; van Oosterom and Meijers, 2011). The two approaches to achieve real-time generalisation exhibit a dichotomy between speed and flexibility. Generalisation algorithms that rely on fast algorithms typically sacrifice cartographic quality to reduce computational complexity and to achieve speed. Approaches for real-time generalisation based on pre-computation and hierarchical data structures lack flexibility and therefore are not capable to adapt to changes in map requirements or content. Detailed reviews of real-time generalisation algorithms can be found (for instance Weibel and Burghardt, 2008; van Oosterom, 2005; van Oosterom and Meijers, 2011; Bereuter and Weibel, 2010).

To master the computationally challenging generalisation process Weibel and Dutton (1999) propose an alternate and complementary approach to the hitherto generalisation models. They differentiate between models with a *process-oriented* and those with a *representation-oriented* view for automated map generalisation. Process-oriented generalisation approaches – the classic generalisation models – are understood to reduce the complexity of a detailed database into a database or a map of reduced complexity, as it is the case of traditional map generalisation. In constrast the representation-oriented view is understood in a complementary fashion to develop consistent multi-scale databases and derive multiple representations. Hence the differentiation between process-oriented and representation-oriented generalisation approaches can be mapped to real-time generalisation relying on fast generalisation algorithms, and real-time generalisation based on pre-computation and hierarchical data structures.

A further strategy has been proposed (Cecconi, 2003) as a combination of both of the above mentioned process-oriented and representation-oriented strategies: the *derivation-oriented view*. The derivation-oriented view derives the subsequent scales from one base, based on a generalisation process and the building of a hierarchical structure. This results in a consistent dataset across the map scales.

Cecconi and Galanda (2002) present an approach for web mapping which extracts level of details (LOD) from an MRDB for computationally expensive feature classes and applies real-time generalisation for those feature classes which can be generalized with less computational effort.

In similarity to the work by (Cecconi, 2003), this thesis aims to bridge the dichotomy between fast generalisation algorithms (process-oriented) approaches and generalisation based on pre-computation and hierarchical data structures (representation-oriented).

Real-time Generalisation, key insights

- The representation-oriented and derivation-oriented view are an alternative to the hitherto process-oriented view to master the generalisation process.
- Real-time generalisation is typically achieved by relying either on pre-computation or on fast generalisation algorithms.
- Real-time generalisation of point features has received only limited attention.

2.4.2. Real-time generalisation based on efficient algorithms

Suitable candidates for real-time generalisation are generalisation algorithms that run in linear or logarithmic time. Among these algorithms, rather straightforward generalisation operators, like selection and simplification are common. Examples include algorithms such as the *Horton stream ordering scheme* (Rusak Mazur and Castner, 1990) or *feature-based global selection* governed by the Radical Law (Töpfer and Pillewizer, 1966). Edwardes et al. (2005), on the other hand apply a local priority criteria to selection operators triggered by spatial conflicts.

Glover and Mackaness (1999) present a system for dynamic generalisation from a single detailed database applicable for urban map generalisation, which combines an automated symbolisation and generalisation technique based on a simple rule base and the application of scale bands.

An approach based on ordered attributes within the GiMoDig project implementing selection, simplification and aggregation operators presents the use of Extensible Stylesheet Language Transformation (XSLT) as a mean to generalize XML-encoded data in real-time (Lehto and Kilpeläinen, 2001; Lehto and Sarjakoski, 2005). Selected line simplification algorithms (Lang, 1969; Douglas and Peucker, 1973) proved sufficiently fast for real-time generalisation, as shown in the theis application in the case of the GiMoDig project (Sarjakoski and Sarjakoski, 2005). This project involves a mobile GI service matching the delivered map types to the user context and preference. It implements the following, well-known generalisation algorithms as web services; feature selection by object class, area and line selection by minimum and maximum extent, contour line selection by interval, line simplification and smoothing and building outline simplification. Both systems implement simple, but efficient, known generalisation algorithms, which were not explicitly developed for potentially huge datasets.

Symbol and label placement algorithms share similar principles with selection and displacement operators. An efficient and scalable algorithm based on a trellis strategy with a unique candidate cost analysis shows real-time behaviour (Mote, 2007). For a problem statement and an empirical study of different approaches and heuristics see Christensen et al. (1995). Harrie et al. (2004) present an algorithm for symbol placement, with a potential for real time applications. The algorithm places the icons sequentially and possibly where they hide as little cartographic data as possible, by performing a spiral search on a pre-computed grid.

Yan and Weibel (2008) present a parameter free point cluster generalisation method based on the Voronoi tessellation. The algorithm takes into account metric and topological measures (number of points, importance value, Voronoi neighbours, relative local densities) based on the Voronoi tessellation and employs the Radical Law to limit the number of points. The algorithms performance depends on the Voronoi tessellation and the iterations needed based on the reduction defined by the Radical Law.

Algorithms implementing more complex generalisation operators such as typification and displacement operators typically rely on iterative optimisation techniques (Mackaness et al., 2007) to achieve a high cartographic quality, and hence are often avoided in a real-time environment.

Burghardt and Cecconi (2007) propose a mesh simplification technique for building typification based on Delaunay triangulation. Each edge within the Delaunay triangulation builds up from the building positions and is weighted according to a set of parameters. The algorithm collapses the buildings on the "shortest" edge iteratively and replaces them with a new representative placeholder building, till the total amount of building defined by the Radical Law is reached. Another approach – although not directly applicable for real-time generalisation – for building typification relies on proximity graphs (Regnauld, 2001), which similarly to the Delaunay triangulation inherently conveys the characteristics of the original point distribution. Based on criteria from gestalt theory the proximity graph is segmented accordingly, with the aim of preserving mean size, shape of the building groups and density.

Mackaness and Purves (2001) present an iterative point displacement algorithm based on the metaphor of social relations to generate radial displacements. The algorithm considers, for each object in turn, a number of close alternate positions and selects the location with the minimal overlap among neighbouring objects. The algorithm shows good performance without the overhead of a spatial tessellation and works well for buildings with a high length-to-area ratio. Lonergan and Jones (2001) present an alternative approach of a displacement algorithm considering alternate position, limiting the overall movement to resolve spatial conflicts.

Real-time generalisation and efficient algorithms, key insights

- Efficient algorithms typically rely on rather simple but effective algorithms and heuristics.

- Fast map generalisation algorithms sacrifice cartographic quality to reduce computational complexity.

2.4.3. Real-time generalisation based on pre-computation

Approaches for real-time generalisation based on pre-computation are often based on dynamic or static tree data structures, to facilitate real-time generalisation (van Oosterom, 2005; Haunert and Sester, 2005; Meijers and van Oosterom, 2011). Hierarchical data structures lend themselves naturally to storing generalisation results and exploiting them for real-time generalisation solutions. Among the authors Van Oosterom is certainly one of the most influential. Over the years he has developed several hierarchical data structures for lines, polylines and polygons, reviewed in Van Oosterom (2005).

Binary Line Generalisation (BLG) is restricted to single line objects and pre-computes the order of elimination based on the Douglas and Peucker (1973) algorithm. The reactive tree for storing polylines is an extension of the r-tree (Guttman, 1984) and stores the importance level of each map object in the tree, which allows the indexing of multiple map objects. Reactive and BLG tree, however, are not suitable for polygon generalisation (van Oosterom, 2005). The Generalized Area Partitioning (GAP) tree considers the special case of polygons (van Oosterom and Schenkelaars, 1995). The topological GAP tree then combines the use of the BLG and the Reactive tree, and avoids problems of sliver polygons along the boundaries on neighbouring polygons by avoiding redundancies (van Oosterom, 2005). Van Oosterom and Meijers (2011) extend the previous tGAP solution and introduce a true vario-scale structure smooth tGAP, by creating a space-scale cube (ssc), which allows for continuous and smooth transition between the map scales.

Hierarchical data structures such as k-d trees, quadtrees or r-trees are good candidates to support generalisation operators. The hierarchical subdivision of space, enables it's indexing by location and resolution, and the efficient finding of proximity relations and scale-related spatial conflicts. Spatial subdivisions such as Pyramids and quadtrees (Samet 1990) are a common approach in GIS.

In the context of line simplification and line filtering Dutton (1999a; 1999b) developed a space-efficient, quadtree-like encoding scheme for positions on the globe called the quaternary triangular mesh (QTM). The QTM is defined in the geographical space and subsequently subdivides each region of the earth into four triangular facets. Each level of generalisation is linked to the resolution of the map and the hierarchical encoding. Line generalisation based on QTM depends on the local position of each vertex in the QTM and does not consider global gestalt criteria, as opposed to other generalisation approaches. The efficiency of this space-based approach (Dutton, 1997) lies within the fact that proximity relations do not have to be modelled explicitly. Such is the case for object-based approaches, where object-primary tessellations such as the Delaunay triangulation are often applied (Weibel and Dutton, 1999) to model and detect spatial configurations.

Sester and Brenner (2004; 2008) present a continuous generalisation approach for mobile mapping application, by subdividing the generalisation process into atomic operations storing topologic and geometric changes. They define elementary generalisation operations (EGO), which are composed of a common set of simple operations (SO). They successively restore, from a simple geometry, the full detailed original geometry. The approach is comparable to the approach of Hoppe (1996) using progressive meshes to simplify triangulated surface meshes, and applied to the following generalisation operators: polyline simplification, building simplification, typification and displacement. Abrupt changes – popping effects – can be avoided by animating the simple operations. Follin et al. (2005) present an architecture for a multiple representation database (MRDB) that allows for the progressive transmission of multi-resolution vector data. Cecconi et al. (2002) introduce morphing techniques to handle continuous scale changes between different level of details within a MRDB, while van Kreveld (2001) investigates how the generalisation operator can be applied in a smooth animated manner for continuous zooming, and Nöllenburg et al. (2008) present a dynamic method for morphing polylines.

A partly automated approach, based on the idea of self-generalising objects (SGOs), has been presented by Sabo et al. (2008). It encapsulates geometric patterns by generating a multi-agent real-time map-generalisation system. For several conventional map productions and for web cartography MRDB has been presented and integrated (Devogele et al., 1996; Anders and Bobrich, 2004; Petzold et al., 2006; Viaña et al., 2006; Brewer and Buttenfield, 2007; Buttenfield and Wolf, 2007; Bobzien and Burghardt, 2008).

Few approaches have been presented of MRDB in combination with location-based services (LBS). The GiMoDig project (Sarjakoski and Sarjakoski, 2005) presented a user-centred approach and Jakobsson (2003) provides a geospatial service, serving harmonised large-scale topographic map content to mobile clients. A generalisation tool by Follin et al. (2005) considers the current user activity for the generalisation process.

Data access through web services such as web mapping services, web feature services or web generalisation services, are an alternate access route to map data, compared to a centralised data management by MRDB (Burghardt et al., 2005; Edwardes et al., 2005). Burghardt et al. (2005) present an operational generalisation system and architecture based on web services and a service registry, subdivided in three service categories: support, operator, and processing services. The integration of the different services has been investigated (Neun et al., 2008b) and an automated control of generalisation operators, in conjunction with data enrichment, showed in case studies the first entirely service-based generalisation system (Neun et al., 2008a).

Operational web and mobile mapping services, such as Google Maps[1], Bing Maps[2], or OpenStreetMap (OSM)[3] and associated 'renderers' such as Mapnik[4], provide seemingly 'continuous' zooming and generalisation. Map generalisation, however, does not occur in real-time. Instead, map tiles are styled and pre-generalised offline and cached at many levels of detail. At run-time, map tiles only need to be retrieved from the cache for the proper LOD. The generated maps are developed for a general usage scenario and some map tile renderers – such as the Mapnik renderer for OSM – allow for rule-based style sheets to adapt the rendered map for different usage scenarios. Also, these map services

[1] Google maps: http://maps.google.com, developer API: https://developers.google.com/maps
[2] Bing maps: www.bing.com/maps, developer API: http://www.microsoft.com/maps/developers
[3] OpenStreetMap: http://www.openstreetmap.org, developer API: http://wiki.openstreetmap.org/wiki/API
[4] Mapnik is a open source toolkit for desktop and server-based map rendering. http://mapnik.org

focus on the base map, not the foreground data, where depending on the map usage real-time performance is imperative. The foreground data (e.g. POIs), are usually rendered in real-time by a different service.

Methods for progressive vector transmission share a lot of similarity with real-time generalisation (Bertolotto and Egenhofer, 2001; Sester and Brenner, 2004; Follin et al., 2005; Yang et al., 2007). Especially as the vector map is ideally transmitted by sending the coarse resolution of the data first, and progressively sending more details – less generalised data – until eventually the whole map is transmitted.

MRDBs in conjunction with automated generalisation have been primarily developed for topographic cartography, with well defined map scales and content. Within the domain of location-based services and web- and mobile mapping applications, however, highly heterogeneous datasets are predominant, originating from various data sources with different accuracies, precision and validity within the spatial, temporal and content domain. Therefore new data models are needed to integrate heterogeneous data into MRDBs and to harmonise and pre-generalise this type of information with respect to the information seeking process of a mobile user. Furthermore, methods are needed which dynamically integrate heterogeneous data on the fly in real-time, foreground as well as background datasets, so that the map content can be flexibly and instantaneously adapted to the information request and context of a mobile user.

Real-time generalisation and pre-computation, key insights

- Generalisation based on pre-computation achieve good cartographic results.
- Solutions of this group, as a consequence of pre-computation, lack flexibility.
- MRDB can be usefully exploited in conjunction with automated generalisation operations
- Continuous generalisation solutions aim for seamless scale changes by using animation or morphing techniques
- Web services for mapping or generalisation are an alternative solution to a centralised management by on MRDB.
- Operational web and mapping services rely on pre-computation to provide seemingly 'continuous' zooming behaviour.

2.4.4. Combined approaches for real-time generalisation

Only a few systems have attempted to bridge the dichotomy between pre-computation and real-time approaches in map generalisation. Cecconi et al. (2002) combine the two concepts: pre-computed level of details for object classes which require a high computational cost for their generalisation, and real-time generalisation for those classes that can be generalized by simple algorithms and heuristics.

The presented real-time generalisation methods encompass selection, simplification and symbolization. A later work in line generalisation employs a hybrid vector-raster solution (Mustafa et al., 2006) by using a GPU programming approach, while storing pre-simplified maps of different scales in a hierarchical data tree.

2.5. Real-time point data generalisation

While map generalisation research so far has mainly focused on line generalisation, the generalisation of point features has received only limited attention. McMaster and Shea (1992, page 71) state that it has been proposed that up to 80 percent of information on digital maps consists of lines. Lines however often form the background and reference information against which the thematic map is represented. With the advent of point of interest (POI) data and the predominance of point geometries in web and mobile applications, point features, however, have gained importance.

Generalisation of points, compared to the generalisation of lines, polygons or fields, has the advantage that the generalisation process does not have to deal with the complexity of shapes and forms especially if performance is importance. However, no additional information on the shape or the topology of a point pattern explicitly exist that can be exploited, as is the case with lines and polygons.

The principles for point data generalisation in real-time are the same as described in the previous Section 2.4. Full detail of existing approaches on real-time point data generalisation are given in Section 3.3.1.

Real-time point generalisation, key insights

- Generalisation of point data has only received limited attention in the literature
- Real-time point data generalisation like line or polygon generalisation relies either on fast generalisation algorithms or on pre-computation.

2.6. Alternative approaches deformation of map space

While operators based on the raster models are defined in the generalisation model of McMaster and Shea (1992), approaches based on the deformation of space are only sparsely discussed in the literature of map generalisation (Edwardes et al., 2005).

Besides the above generalisation algorithms that directly manipulate the point objects (e.g. by aggregating or displacing points), alternative approaches have been proposed that manipulate and deform the map space (e.g. fisheye transformations), similar to the approach used in Focus+Context techniques in visualisation (Furnas, 1982). Edwardes et al. (2005) investigate the portrayal of topographic base maps with ad-hoc queries, and lay-over of points from geographic data base, particularly with respect to how the spatial relation between map objects and the background map is maintained. They advocate modelling the distortion caused by the map symbol covering up the underlying space -with a variable scale projection magnifying the neighbourhood of point symbols. The model of the distortion space caused by the symbology can not only be applied as a mean for cartometric analysis, but also as a generalisation operator to reconfigure the symbolized features and therefore to preserve the point set topology of the underlying space.

In the context of mobile mapping a variable scale approach is motivated and promoted (Zipf and Richter, 2002) with the aim of providing the map user with a detailed map of his current location – the focus – and an overview – the context – simultaneously. More details on existing approaches on deformation of maps' space in the context of map generalisation are presented in Section 3.3.2.

Approaches based on the deformation of space, key insights

- Approaches based on the deformation of space are only sparsely discussed in the literature of map generalisation.
- Variable scale maps have been advocated in the context of mobile mapping.

2.7. Research gaps

With the advent of mobile cartography and web 2.0 applications, point data gained importance, a trend which so far has not been reflected in generalisation research. In comparison with line or area generalisation (AGENT, 1998; Stoter et al., 2009), point data generalisation has received relatively little attention.

Many solutions have been conceived, with a rather opportunistic perspective concentrating on rather 'small' generalisation problems. A comprehensive, problem-oriented approach for real-time point data generalisation that uses a common conceptual grounding is absent. Such conceptual frameworks (McMaster, 1987; Plazanet, 1995, 1996) exist for line generalisation, where they have had a very positive effect on the further evolution of research. Furthermore, existing frameworks rarely account for variable scale maps – that is, generalisation approaches which manipulate the map space instead of map objects.

Secondly methods are largely missing that combine the advantages and strengths of both main approaches in real-time generalisation: real-time algorithms for point generalisation exploiting hierarchical spatial data structures. In other words, a generalisation approach that attempts to bridge the existing dichotomy between pre-computation and fast algorithms, between flexibility and speed.

Lastly, solutions are needed which account for the requirements of mobile and web 2.0 applications. Solutions which meet the need and expectations of interactivity, usability and modularity. Solutions which link cartographic generalisation and geographic visualisation.

Chapter 3.

Generalisation of point data – a problem-oriented approach

The literature review showed that a comprehensive, problem-oriented approach that uses a common conceptual grounding for point data is missing, while comparable approaches exist for line generalisation.

There are two motivations for such an approach: firstly, to situate the specific requirements of point generalisation within a generic generalisation framework, and secondly, to account for the specific requirements of mobile- and web mapping, especially in terms of real-time performance and flexibility. This chapter aims to address this research gap and tries to contribute towards developing such a workflow for point generalisation, against the background of information portrayal on mobile devices.

The chapter starts with an analysis of the dimensions that define the point generalisation problem. It then presents a classification of point generalisation algorithms, which is fed into a problem oriented workflow for point generalisation for web and mobile mapping.

3.1. Definition of the point generalisation problem

Generalisation systems have been neither integrated nor explicitly developed for web mapping. Gaffuri (2011) identified the following issues from an architectural point of view: *raster versus vector data* – web map servers typically provide raster data; *time constraint* – real-time requirement on the client side; *integration* – data integration of heterogeneous data available on the internet; *genericity* of the generalisation algorithms to be capable to deal with the large diversity of data; *complexity of generalisation* which may be difficult to use for web developers; the *non availability of automatic web generalisation libraries* for web mapping.

In what follows, point generalisation is deconstructed into its defining factors. Those factors play a significant role in how they constitute and influence point generalisation. These are: the role of background vs. foreground data, the types of point data, constraints that act on point data, and the level of interactivity requested by the application.

3.1.1. Background vs. foreground data

At least in parts, point data typically form, – depending on the type of the map – the focus of the map, and hence represent the foreground. The background data are those acting as a base map, in the traditional cartographic sense. Background data provide a visual backdrop and spatial reference. They conventionally provide the user with hints for orientation and, sufficient reference for the foreground data. However in order to fulfil the role of spatial reference, they also impose spatial constraints on the points in the foreground.

3.1.2. Types of point data

Arguing from a generalisation point of view, point data can be grouped into so called *points of interest* (POI) and *point collections* (PColl) depending on the spatial phenomena that they depict (Burghardt et al., 2004a; Mannes, 2004). These two groups form the two ends of a spectrum. At one end of the spectrum lie points where the absolute position is important such as in the case of POI data and on the other end points, where the relative position within the group – and spatial pattern – is important. In the context of maps, a point is an abstract 'one-dimensional' representation of a human concept or real world phenomenon at a specific location on the map, in the form of a symbol.

Table 3.1.: Qualitative aspects of point data between the two ends of a spectrum between POI and PColl.

Points of interest POI		**Point Collection**
	⟷	
Properties		
Unique identity		Generic
Absolute position		Relative position
Heterogeneous		Homogeneous
Qualitative attributes		Quantitative attributes
Examples		
Restaurants	Weather stations	Animal observations
Mountains	Sensor nodes	Flickr data

POIs are, as the name suggests, of particular interest for a particular application, which also means that they are largely self-standing, existing on their own. Examples of POI include typical landmark features such as restaurants, sports facilities, civic infrastructure etc. – what is labelled an 'amenity' in OpenStreetMap data[1] – or other address layer data. They are often typical landmark features serving in orientation, covering actual physical space, and exhibiting some sort of identity and uniqueness in space. This also implies that POIs are more constrained in terms of spatial relations, such as the topology. The Italian restaurant on the street corner must stay in this particular topological relation within the street network, even after generalisation, as it may also act as a landmark. The attributes of POIs are often rather heterogeneous and of a qualitative nature that is dependent on the type of POI. Conversely, PColl data are representatives of collections of points and may be more liberally selected, aggregated, typified and displaced – as long as the overall spa-

[1] OpenStreetMap (http://www.openstreetmap.org) is a collaborative project to create and distribute free geographic data for the world, initiated in UK in 2004.

tial distribution of the point set is maintained. Examples of point collections encompass any point data that exist in large collections, such as count data or categorical observations collected at point locations, (e.g. animal observations, etc.). Finally, POIs typically have rich qualitative attributes associated with them, while PColls typically don't, or feature rather quantitative attributes. Examples of a mixture of both types include fixed measurement stations – such as weather stations – with rich quantitative attributes associated (like wind speed or temperature).

3.1.3. Constraints on point data

Different type of constraints can be formulated to constitute the point generalisation process. Aside from cartographic and foreground constraints, background-foreground constraints are present, to account for the often dynamic on-demand nature of the foreground data.

Background-foreground constraints As mentioned above, the point data in the foreground are subject to spatial constraints by the background, or base map elements. Examples include topological relations, such as a landmark POI which should not switch sides of a road, or a watershed which may forms a 'container' to mass points collected for that watershed. Besides topological constraints, the background also imposes other constraints on the point data, relating to the other two types of spatial relations commonly distinguished: proximal relations and directional relations (Jones, 1997). Furthermore, constraints regarding the readability of point symbols ensure the visual separation between foreground and background features and facilitate the identification of foreground features.

Foreground-foreground constraints Constraints also act between the foreground objects themselves. If dealing with point objects, topological relations would seem irrelevant. Yet they are relevant, since points on a map are small area objects, owing to the fact that they are represented by cartographic symbols. And obviously, proximal and directional relations are also relevant.

Cartographic constraints Maintaining the above spatial relations is one of the key mandates of any generalisation activity, and therefore part of the standard set of cartographic constraints. Generalisation, however, also has to observe additional cartographic constraints relating to the structure, pattern, and semantics of point sets, such as density, spatial arrangement, or relevance ordering. Also, constraints that ensure readability and identification of foreground map features are of importance, ensuring the minimal symbol size and separability between point symbols. Several typologies of cartographic constraints have been offered in the literature (cf. Section 2.2, e.g. Steiniger and Weibel, 2007).

3.1.4. Level of interactivity

Unlike in paper map production, it can be safely assumed that the user has the possibility of interactively adjusting the map display. A user is likely to expect the system to deliver this capability through various interaction modes such as mouse, touch, voice or gestures. Common interaction features are; zooming, panning, changing the representation, switching feature classes and map layers on and off and interaction with single map elements.

It can be said that the higher the interactivity, the more the generalisation quality can be relaxed and left suboptimal, such as resolving visual clutter by zooming in. Still, in order to know where to 'zoom in' relevant features on a global scale have to be retained. Finally a higher degree of interactivity raises expectations of the user regarding response time and real-time performance.

3.2. A conceptual classification of point generalisation algorithms

3.2.1. Point generalisation

Point generalisation algorithms may be organised by various criteria. From a generalisation perspective, it makes sense to base the classification on how map space and map features are affected by the algorithms; essentially, in how the visual representation and information is affected. Based upon these criteria the geometric algorithms for point generalisation are organised into a hierarchical, three-level classification (Figure 3.1).

Maps convey information by the means of symbols and a pictorial representation of space, typically on a planar medium (Berendt et al., 1998). Thus the first level distinguishes the transformation focus and states how the algorithm acts. The transformation can either focus on the map objects – thus being *object-directed* or concentrate on a transformation of the map space, with a *space-directed* focus. For wide scale ranges, instead of mere generalisation, there is also the option of a change of the map representations method, which changes the representation mode of the map data. Especially over large scale ranges a change of representation mode is favourable over a plain generalisation of map data.

At the second level the classification differentiates based on the principle used to execute the generalisation operation. The third level lists the actual generalisation operations, for object- and space-directed transformation foci. The notion of generalisation operators for object-directed generalisation operators here follows the understanding of Weibel and Dutton (1999) – where generalisation operators are independent of the applied data model by the generalisation algorithm – in contrary to McMaster and Shea (1992), who differentiate between location- versus object-based operators. In effect they distinguish between raster- and vector-based generalisation. Space-directed operators are organised according to the applied distortion operation and respective algorithms. This thesis follows the definition of algorithm by (AGENT, 1999). An algorithm, denotes a formal mathematical construct that solves a generalisation problem by changing an object's geometry or attribute (AGENT, 1999).

Situating the algorithms within the classification provides a good base for the generation of an overall workflow and system for point data generalisation, and aligns them with the specific constraints (see above) for real-time point data generalisation. Based on an extensive literature review, a selection of exemplary point generalisation algorithms, with a high potential for real-time generalisation are presented to illustrate the classification of point generalisation algorithms. The selection favours simple and fast methods and those

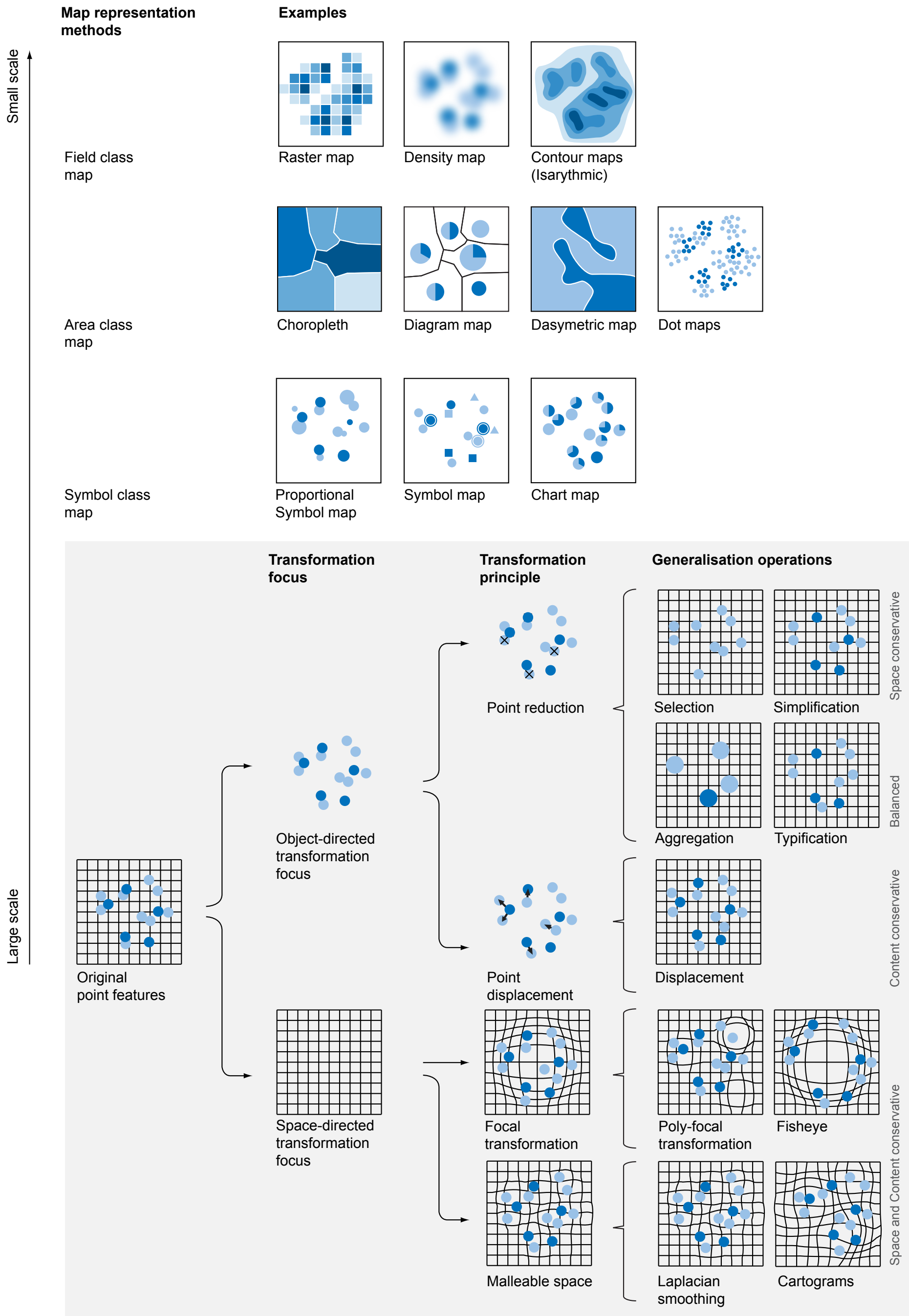


Figure 3.1.: Methods and operators for point generalisation on mobile devices for 2D map representations.

based on pre-computation. Algorithms relying on extensive spatial analysis and iterative optimisation are thus ruled out. It has to be said, though, that for small numbers of point objects optimisation techniques (e.g. for point feature displacement) may well be feasible even today, and even more so in the future.

3.2.2. Map representation methods

In map design, representation methods are selected according to the data available, the map purpose and scale. The same criteria hold for map generalisation. Especially for generalisation over a wide range of scale, the map representation methods need to be considered and integrated into the overall generalisation workflow. In the case of point generalisation – and depending on the map purpose – points may be aggregated to collections, areas or fields, which in the case of areas and fields alters the dimensionality of the data representation. For instance, for large changes of scale, point features may be transformed into density fields, highlighting the spatial aspect (Berendt et al., 1998). A Kernel Density Estimation (KDE) may be applied, or points are clustered into groups of polygons emphasizing the symbolic aspect.

On the example of the generalisation of settlements, Ratajski (1967) defined control points at which scale the representation methods for map features change, and defined scale ranges within which a representation method is valid. In web cartography Cecconi and Galanda (2002) took a similar approach by defining a limit of applicability for different object classes. Müller et al. (1995), more dramatically, called the large variation in graphic representation of objects in map series *catastrophic levels*. For instance, if between two consecutive scales the representation of a group of single houses turns into a polygon envelope, representing a conceptual change in representation.

On a qualitative scale, from large to small scale, symbol-, area- and field-based representation methods are the three main representation methods applicable to point data generalisation. Symbolisation and classification depends on the map data and purpose, and is defined during the map generation process. However in the case of interactive maps, symbolisation and classification may be changed on-the fly by the user. Symbol class maps, with symbol representation methods, are applicable for rather large scales, such as the application of different symbol types to categorical data or various types of diagram maps for aggregated data. For medium scale ranges, large point data sets may instead be qualitatively aggregated to areas rather than reduced to one aggregated point.

Thematic map representation methods for area class maps encompass *choropleth maps*, *area diagram maps*, *dasymetric maps* and *dot maps*. Area class maps provide an aggregated overview of the source point data and change the dimensionality of source data. That is, they combine a set of objects into one object of a higher dimensionality equivalent to the combine operator (AGENT, 1999). *Raster maps*, *density maps* (e.g. kernel density maps) or *isarythmic maps* (contour maps), aggregate the source point dataset to field class maps. Field-based representation methods are especially applicable at small scales, as they provide a good insight of the spatial distribution of the source data.

3.3. Algorithms for point generalisation

3.3.1. Algorithms with an object-directed transformation focus

Algorithms with an object-directed transformation focus modify point objects without changing the metrical properties of the underlying spatial projection. This broad class represents the generalisation operations in the strict sense, as termed in the generalisation literature (see Section 2.3). The generalisation operators which apply to point generalisation in the classical sense are (Edwardes et al., 2005): *selection*, *simplification*, *typification* and *aggregation*, that reduce the number of features represented on the map. The displacement operator moves the map features away from each other to remove overlaps and congestion. Depending on the generalisation process, generalisation operators are applied either locally in a specific map region or on a global level on the whole map. The working transformation principle used by these operations, to achieve the generalisation transformation, is either point reduction or point displacement; in other words, operators affecting the quantity or the cartographic quality of visual information similar to the Ratajski model (Ratajski, 1967, after McMaster 1991). In contrary to the distinction by McMaster and Shea (1992), based on the underlying data model, this principles distinguishes between spatial and attribute (semantic) transformation . For each of these generalisation operators different algorithms are conceivable, some of which already exist in the literature (see Chapter 2).

Note that for both groups and corresponding algorithms to implement the generalisation operators, global and local application is possible. It depends on how the operators are combined and on the overall implemented generalisation process.

If all types of map objects were included into this classification (point, lines, areas), the following operators would include, for single objects, simplification and enhancement of shapes and lines, and for groups of objects, amalgamation as well as aggregation for polygons.

3.3.1.1. Quantitative point reduction

Point reduction algorithms reduce the number of points represented on a map as a function of the point density and the scale reduction factor applied. The assumption is that excess, unimportant points exist that need to be removed in order to make room on the map.

Selection and simplification constitute a true subset of the original points and follow by leaving the point location unmodified; a space-conservative approach. The more complex generalisation operators – aggregation and typification – generate replacement features of the original points. These are not necessarily represented at their true location and therefore follow a balanced approach between reduction and displacement.

Selection Selection is a subset of points from the original set of points, based solely on attribute values, such as a relevance value per point object. Selected points remain at their original positions. The resulting number of points can be obtained from the Radical Law (Töpfer and Pillewizer, 1966). Globally, selection is equivalent to a (simple and fast) filtering operation, primarily used to create a candidate set of points potentially involving conflicts that must be solved by other generalisation operations. Locally, selection may be used to select the more important of two overlapping point features (Edwardes et al., 2005), or based on a density criteria of quadtrees (and octrees) (Peters, 2013).

Simplification Like selection, simplification chooses a subset of the original points that remain at their original positions. However, it is governed by geometric properties such as proximity or density of points. The purpose of this operator is usually to relax the solution space for the conflicts rather than solve them entirely. The most trivial simplification algorithm would simply select points randomly. De Berg et al. (2004) describe algorithms for simplifying point patterns using epsilon-approximations (squares, rectangles), that aim to preserve density variation across the map space. Rather than using a uniform grid, hierarchical data structures such as quadtrees might be used that adapt to density variations (similar to Burghardt et al. (2004b), who used quadtrees for aggregation).

Aggregation Aggregation denotes the replacement of two or more point features with a new placeholder feature. Its main purpose is to reduce the level of detail in the map by grouping together semantically similar and spatially close points. In contrast to selection and simplification, aggregation generates new point locations. Aggregation may be implemented in two ways. First, by clustering algorithms (see Anders (2003) for a review), where hierarchical methods provide the possibility for precomputation (Mannes, 2004). Second, by hierarchical data structures including k-d trees, quadtrees (Burghardt et al., 2004b) or reactive trees (van Oosterom, 1992).

Typification Typification can be seen as a type of aggregation and is one of the more complicated generalisation operators. However, it differs in that it uses the pattern of spatial relationships among a group of points to imply the existence of a new pattern, that aims to carry the characteristics of the original point distribution. In order to get a grasp of the arrangement of point patterns, Delaunay triangulations (Burghardt and Cecconi, 2007) or proximity graphs (Regnauld, 2001) have been used. Sester and Brenner (2004) provide an approach for continuous generalisation.

3.3.1.2. Qualitative point displacement

As a qualitative operator displacement works locally, it reconfigures point symbols in order to resolve conflicts, by moving them apart, leaving their number intact. Displacement is usually applied for the 'finishing touches', as the last in the chain of generalisation operators. Many displacement algorithms exist in the literature, but most rely on computationally demanding optimisation techniques and/or geometrically complex spatial analysis. Hence, only a few algorithms remain that can potentially operate in a real-time environment, including an algorithm using the least disturbing space (Harrie et al., 2004), or metaphors of social relations to generate radial displacements (Mackaness and Purves, 2001). The greedy algorithm for energy-minimising 'snakes' may also be a possibility (Kass et al., 1987). Another very simple algorithm could be envisioned that would displace points to regularised grid locations. Given that mobile services are inherently interactive, interaction techniques, such as local radial displacement induced by mouse-over events provide an alternative approach to displacement.

3.3.2. Approaches based on a space-directed transformation focus – variable scale maps

Monmonier (1977) suggested nonlinear reprojection to reduce clutter of thematic map symbols. For small-display cartography Harrie et al. (2002b) suggested the use of variable scale maps to present geodata to the mobile user. One of the key aims of map generalisation is to reduce spatial conflict, such as overlaps or congestion between map symbols. In principle, this can be achieved in two ways. Classical generalisation operators transform the map objects, keeping the map space fixed. However, conflict can also be reduced by a space-directed transformation, namely transforming the map space, while map objects are moved apart as a result of space deformation. The latter approach is a direct counterpart of feature displacement, as spatial conflicts are resolved by deforming the underlying space, the overall map surface. Map space does not necessarily have to be Euclidean, the representation presented on the screen, however, is.

Two transformation principles may be applied: focal projections and the idea of a malleable space.

3.3.2.1. Focal projections

The group of focal projections deforms the underlying map space at one or more predefined regions. Focal projections encompass focus+context interaction techniques in graphic visualisation, based on the metaphor of lenses. The predefined regions of interests are typically circular regions, which are deformed radially from the focal centre, creating a lens effect.

Various and more generic solutions for one focal centre (Snyder, 1987; Sarkar and Brown, 1994; Fairbairn and Taylor, 1995) and polyfocal projection (Kadmon and Shlomi, 1978; Carpendale, 2001; Pindat et al., 2012) exist. While some fisheye views exhibit large distortions, Yamamoto et al. (2009) suggest the inclusion of a glue area between focus and the overview area absorbing all the distortions of the map and leaving the rest of the map undistorted. Implementations of fisheye views (Furnas, 1982) with one focal centre for small display maps have been shown by Harrie et al. (2002a) and Rappo et al. (2004).

3.3.2.2. Malleable space

The second principle adapts the space deformation locally to the existing map objects, following the idea of a malleable space.

Laplacian smoothing Edwardes et al. (2005; 2007) investigates Laplacian smoothing to reorganise points and relax spatial conflicts. Laplacian smoothing was originally used in computer graphics to smooth the representation of surfaces. The implemented approach based on a primal-dual scheme Taubin (2001) proved sufficiently fast for interactive portrayal and was found suitable for contemporary mobile devices.

Cartograms Cartograms and various implementations thereof, can be ascribed to the same category. Cartograms are also known as anamorphic, or density-equalizing, maps. Tobler (1973) described cartograms visually by using the metaphor of a stretchable 'rubber sheet'. He then proposed cartograms as a means of creating voting district boundaries and stated: 'Imagine that one could stretch a geographical map so that areas with many people would appear large, and areas with few people would appear small'. Several implementations exist; the implementation of Dougenik et al. (1985) and the extension by Sun (2013) are based on the idea of force fields. Whereas the first approach is based on polygons, the latter implementation approximates area polygon efficiently by squares or circles, and preserves the topology. The algorithm by Gastner and Newman (2004) is founded on the principle of diffusion, known from particle physics. A solution with the focus on large point sets is presented by Bak et al. (2010), the result, however, highly depends on the source dataset and the amount of selected centres.

A key advantage of space-directed approaches is that both foreground and background objects are displaced together, thus maintaining spatial relations. However, for both space-directed transformations, the perceptual and cognitive validity remains to be evaluated (Carpendale, 2001; Mountjoy, 2001; Lam et al., 2007; Schafer and Bowman, 2003; Pietriga et al., 2010).

3.4. A problem oriented workflow for map-generalisation

3.4.1. Workflow overview

Mackaness (2006) stated that automated cartography concentrated on mimicking traditional cartographic map making and that 'there has been too little concern for cartographic methods that go beyond what was possible on static printed maps'. The focus of mobile and web maps shifts toward sustaining more interactive and dynamic interactions, owing to new interaction technologies such as touch screens. The proposed problem-oriented workflow for point generalisation addresses this shift, by taking into account the dynamic and real-time nature of mobile environments by considering the factors laid out in the beginning of this chapter. The workflow summarised in Figure 3.2 includes resource definition, map purpose, transformation focus, generalisation strategy and generalisation process and is explained in detail below.

3.4.2. Resource definition

In the first step of the workflow the available resources are defined. These are not assumed to change throughout the generalisation process. Resources have an impact on the generalisation process, changing cartographic constraints (e.g. minimum symbol size for interaction on touch screens) and the generalisation strategy applied. Resources include the mobile device (i.e. the client), background processing capabilities, such as server-side services, and the constraints of the human sensory system. Relevant resources for the mobile device are screen resolution, processing power, local storage capacity, network bandwidth, and user interface and interaction capabilities.

3.4.3. Map purpose

The second step sets the purpose of the map. The purpose of the map informs how the map should reflect information throughout the various scales, which transformation focus is used, and the generalisation strategy applied. Mackaness (2006) suggested moving

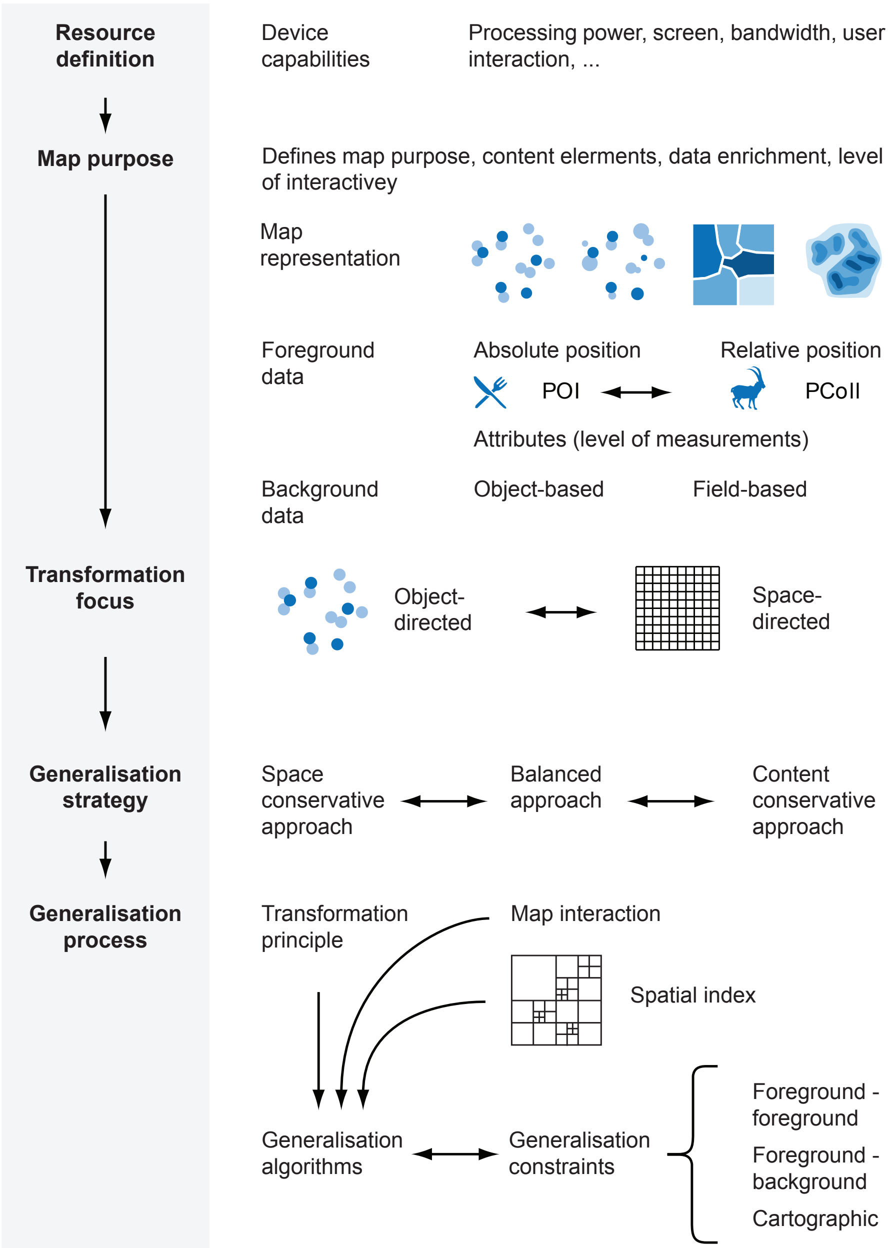


Figure 3.2.: Workflow for mobile point generalisation.

from automated cartography and the respective generalisation paradigm of "capture once, use many times" to an improved inclusion of meaning and context, and the aspect of exploration in maps. He assigned interactions a significant role in how humans give meaning to things, and postulated a need to identify the optimum balance of decision making between the human and the system.

This step selects map representation, symbol types, the content elements, the background and foreground of the map, level of interactivity and the degree of pre-filtering or enrichment of the data. Thus what type of map visualisation is used, and the message the map should carry, has an effect not only on the visual appearance of the map, but also defines to a certain extent how the generalisation process is conducted.

Different elements play a role in defining the purpose of a map: the available resources, the user, the given task (*e.g.* planning, orientation or exploration), and the mobile context. These elements, after all, affect point generalisation, and hence the type of foreground point data (POI or PColl), the map type, the importance of background or reference data, the constraints on point data, and the level of interactivity.

The definition of the map purpose and ongoing steps in the workflow depend on the requirements of a particular user, and will therefore change and/or re-run depending on interaction capabilities, as well as the user and map purposes.

3.4.4. Transformation focus

The transformation focus defines whether to select an approach based on an object-directed or space-directed transformation focus. The applied algorithms solve spatial conflicts by either deforming the map space or transforming feature points on the map space.

An example of using a space-directed approach would be a map for a tourist wandering around Zurich. Here the relative positions of landmarks immediately around him/her are of greater importance than unreachable ones. On the other hand, if the very same tourist requests a map to understand the spatial distribution of the fox population in the City of Zurich[2], an object-directed transformation focus with a content-conservative generalisation strategy, may be of better use.

Note that the different transformation foci and principles are not mutually exclusive; they merely represent solutions to the same problem (unambiguously placing map symbols on scarce map space) and can possibly be mixed.

3.4.5. Generalisation strategy

The strategy for point generalisation formulates generalisation constraints and defines how spatial conflicts are resolved during the generalisation process. The content conservative approach tries to retain as many point features as possible on the map, and prioritises displacement as a generalisation operator. It assumes that the point features have been previously filtered to a sufficiently small number.

[2] There are indeed a great number of 'city foxes' in Zurich. There are estimates of approximatively 10 foxes per square kilometre for the city of Zurich. See: http://www.bafu.admin.ch/tiere/09262/09313

The balanced approach resolves spatial conflicts by aggregating point features, and is better suited for highly interactive maps that need a larger 'interaction footprint' per point feature. This approaches prioritises typification and aggregation operators.

The space conservative approach tries to avoid displacement of point features and prioritises selection and typification as generalisation operators. The space-directed transformation focus is itself a content conservative strategy, as it retains the foreground data, and is space conservative as foreground and background data are transformed together. However if after/before a successful deformation object-directed transformations are applied, the strategy depends on the generalisation operators applied.

3.4.6. Generalisation process

How the generalisation process is executed is driven by the generalisation strategy. The generalisation process includes the selection of algorithms (real-time or pre-computed data structures) and the chaining and execution of single generalisation operations (see Figure 3.1), which ultimately enables the map generation and visualisation on the device. The process is controlled by the generalisation constraints (see Section 3.1.3) in order to meet cartographic legibility constraints. Map interactions such as zooming, panning, or the selection of details cause a change or a rerun of the generalisation process.

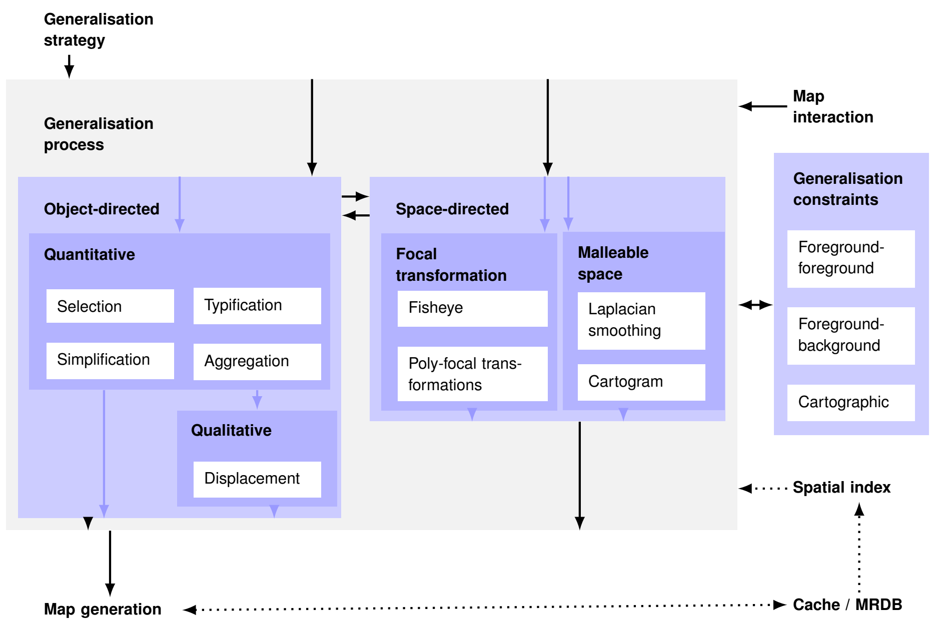


Figure 3.3.: Generalisation process highlighting possible interaction between transformation principles

The generalisation process itself is built in a modular manner. It allows chaining of the different transformation foci and principles on a global- or local level of the map. The modularity provides a flexible way to execute the generalisation process, especially as the application scope of generalisation operators changes over the range of scales (Cecconi et al., 2002).

For instance a plain object-directed process applies selection and, if required, adjusts eventual remaining cartographic constraints with a displacement operator. A plain space-directed process would execute one of the space-directed approaches. The combination of both transformation foci is possible, and used to remove additionally introduced congestions by the spatial deformation of the space-directed approach. In order to already leverage congestions prior to the spatial deformation and keep enough detail on emphasized regions local reduction of detail is applied accordingly (see Harrie et al., 2002b; Rappo et al., 2004; Meijers, 2011) – termed – in the case of focal transformations – *radial generalisation* by (Reichenbacher, 2004).

The generalisation process optionally supports the generalisation algorithms, with a spatial index and a cache of previous queries to speed up the overall generalisation and provide optimal map interaction.

Spatial indices which are optimized to efficiently handle geometric queries developed since the early days of geographic information systems GIS (for an early overview see Peuquet, 1984), are applicable to leverage the dichotomy between pre-computed and fast but straight-forward generalisation algorithms (see Chapter 2). This dichotomy is partly founded on computationally rather expensive nearest neighbour queries, leading to a computational bottleneck during the generalisation process.

Various spatial indices exist such as grids, quadtrees, R-trees or KD-trees and are extensively applied in various fields where spatial queries are of importance (Kresse and Danko, 2012). Among these the quadtree has been widely suggested for GIS (Jones and Abraham, 1987; Gahegan, 1989; Samet, 1989). The quadtree as a spatial index comes with a set of properties (see Section 6.1), which makes the quadtree a natural candidate as a support index for real-time point generalisation. A spatial index should not only allow possible hierarchical storage, but furthermore support the algorithms to preserve topology, shape, visual hierarchies, maintain proportions of classes, allow for displacement, simplification, and satisfy external constraints.

This work investigates a modular set of quadtree-based algorithms and their application for real-time generalisation. The inclusion of a hierarchical spatial index permits the bridging of the dichotomy between the two real-time generalisation approaches and the profiting from the resulting flexibility and speed of the approach.

3.5. Concluding remarks

This chapter delivered three things: a definition of point generalisation in the context of web and mobile mapping; an extension of the typology for point generalisation operators and respective algorithms; and the proposal of a workflow for point generalisation in web- and mobile environments.

The extension of the typology for generalisation operators (McMaster and Shea, 1992; AGENT, 1998) introduces a space-directed transformation focus to the existing object-directed approaches. It also links two aspects of how generalisation can be looked at, from a strictly object-directed or space-directed point of view. Additionally, it allows integrating algorithms that deal with spatial transformation into the generalisation process.

This distinction feeds into a workflow for mobile point generalisation that furthermore considers the trade-offs between content- and space-conservative strategies for generalisation, defined by the map purpose. Given a clear map purpose, the transformation focus and generalisation strategies can be defined and conducted in the generalisation process.

Finally, the generalisation process integrates modularly the different generalisation operations, based on the generalisation strategy, constraints and map interaction. Furthermore, the integration of spatial index structures is highlighted and will be elaborated further, based on the quadtree, in the coming chapters.

Chapter 4.

Content zooming and information exploration for web and mobile maps

4.1. Interaction and map generalisation

One of the key challenges in comparison with traditional paper maps stems from the limitation of the screen size, especially for the display of overview information. Following Shneiderman´s *Visual Information Seeking Mantra* (Shneiderman, 1996) "overview first, zoom and filter, then details-on-demand", the starting point of an orientation and navigation task is typically the overview perspective. Only then is additional and more detailed information displayed for specific areas in an automated or interactive way.

In a mobile map application, the user is confronted with different tasks during the exploration of space related problems. He/she has to get a grasp of what can be found on the map and where it is located. The question of where is primarily answered with the help of the base map (or background map), which provides sufficient spatial reference for the foreground objects, such as points of interest (POI), highlighted routes or selected areas. The background map is, in the first instance, independent from the usage scenario of the map application. Question on the what is focused on in the foreground. Conventionally, the assumption in map generalisation is that the level of detail (LOD) of the map background and foreground should always correspond, and thus change synchronously across scales. However, depending on the usage scenario, mobile users may want to override the rules of classical map generalisation, adapting the content representation to the given information seeking task.

This chapter proposes a methodology for exploring spatial information, that is based on the principle of decoupling spatial zooming and content zooming. Spatial zooming refers to the change in map scale by zooming in or out on a map (i.e. a mere scaling operation). Content zooming denotes the operation of changing the detail of the content of a map, for a given map scale. Conversely, in classical map portrayal, cartographic map generalisation (called standard zooming here) always performs both content zooming and spatial zooming. In other words, changes to the map content are always a function of the map scale. In addition to standard zooming, in this methodology tools are made available that enable content zooming and thus support an individual adaptation of foreground data, allowing the overriding of the effects of standard zooming. That is, the user can decide how much (foreground) content will be shown, and in how much detail it is represented, independently of the selected map scale and map extent.

This chapter first provides an overview on visualisation and interaction techniques, and cognitive research approaches. Then it introduces the methodology for content zooming and information exploration. The methodology is implemented within the framework of a research prototype that offers a wide variety of quadtree-based algorithms for real-time generalisation of point data, see Chapter 6. Appendix A provides an in-depth description of the prototype. Two case studies present the methodology in use, and show how entire workflows for information exploration can be built, chaining together different operations of generalisation and content zooming. The chapter concludes after a discussion, with some general insights and an outlook on future research.

4.1.1. Visualisation and interaction techniques

In LBS the portrayal of map content is limited to a small screen. Geovisualisation deals with the question of how to represent data, particularly large amounts of data in a computationally efficient and visually and cognitively effective way. To provide the user with the capability of interacting with the map, in spite of a restricted screen real estate, different approaches have been suggested in the LBS and human-computer interaction (HCI) literature.

Shneiderman (1996) proposed a guideline for visual design quoted in the introduction of this chapter. He states that for a visual design to be successful, the interface should support the following tasks: overview, zoom, filter, details-on-demand, relate, history, and extract. In the case of LBS most of the tasks are incorporated, especially panning and zooming.

Map Content – for the selected map extent – is often too voluminous to be visualised with the desired degree of detail on a single screen, which is especially true for maps on a mobile device. This raises a challenge which cannot solely be solved by providing appropriate generalisation of the map content: How can both context and detail be integrated simultaneously into the portrayal of map content?

In the literature on 'focus-plus-context' (also known as detail-in-context), this question is addressed and different solutions are provided. The basic idea of focus+context techniques is to show selected regions of interest in greater detail, while preserving the overall context, avoiding occlusion. In a survey on different focus+context solutions, Cockburn et al. (2006) identify four main mechanisms employed:

1. *spatial separation* applying two separate views for overview and detail information
2. *temporal separation* applied in zooming to separate two spatial resolutions in time
3. *cue-based separation* to highlight focal objects
4. *seamless focus+context views* with focus+context within a continuous display, for instance distortion-oriented visualisation like the fish-eye view

Several distortion-oriented visualisations have been proposed for mapping applications. One of the earliest known focus+context visualisations is the fisheye view proposed by Furnas (1982). Zipf and Richter (2002) promoted focus maps for LBS to ease map reading, stating that focus+context representations ease map reading in that they focus a user's attention on the area of a map that is of interest to him/her. Several focus+context visualisation techniques were proposed for LBS, such as (Harrie et al., 2002a; Pindat et al., 2012; Yamamoto et al., 2009). Keim et al. (2009) present focus+context displays in a different context, for scatter plots.

Nevertheless, focusing in on an area of interest, showing it at greater zoom level invariably causes distortion, making the visualisation not necessarily easy to interpret and navigate (Carpendale et al., 1997; Pietriga et al., 2010).

4.1.2. Cognitive and user oriented approaches

Motivated by work on map usability (Winter and Tomko, 2004; Nivala et al., 2007) Cheung et al. (2009) propose a cognition-based approach, positing that the more zoom levels (or LODs) a map has, the more inconvenient and complex map reading becomes. They propose, similarly to Robbins et al. (2004), a reduction in the number of LODs for the user to receive the required information. Furthermore, they implemented progressive visualisation that asynchronously provides further details on a specific map scale in an attempt to ease map reading.

Assigning relevance to geographic objects in LBS is addressed by Reichenbacher and De Sabbata (2011), who define geographic relevance (GR) as a quality – the relevance – of an entity in geographic space and a given context that extends beyond location (e.g. user characteristics). As geographic entities are situated in space, they are spatially organised and related to one another. Therefore, De Sabbata and Reichenbacher (2012) argue and demonstrate that users do consider spatial relations between objects such as co-location as being important. Here relevance is assumed to be 'static' and furnished as input for a particular query by an external relevance ranking algorithm.

4.1.3. What is missing?

In summary, what is still missing is the capability for the user to adapt the portrayal of the content to his/her needs, overriding the proposed 'standard' generalisation solution to increas or reduce the amount of foreground objects independently of map scale, and thus get better support in solving his/her information seeking task. Approaches that link human computer interaction and automated generalisation assume that interaction plays an important role in how humans give meanings to things, and that cartographic decision making may not entirely be shifted to the mapping system (Mackaness, 2006; Weibel, 1991). In the following, we propose information exploration through content zooming as a methodology for providing this desired functionality. The methodology shows how interaction can be flexibly integrated with existing automated generalisation strategies, which differ in the complexity of the generalisation operations employed. Thus, the possibility of adjusting the degree of detail and adapting the generalisation outcome is shifted to the user.

4.2. A methodology for information exploration by content zooming

Content zooming varies the degree of detail on a map without changing the map extent. It facilitates changing the degree of abstraction on a map independently of its, scale in contrast to cartographic map generalisation, which acts as a function of map scale.

Content zooming implements two aspects: the change of the amount of foreground objects, that is, the number of foreground objects displayed for a given LOD; and the change of the granularity (see Section 4.2.3), referring to how detailed / dense the represented information is, in spatial and thematic terms. This relates to filtering and details-on-demand in Shneiderman (1996).

The presented methodology consists of three steps, with an increasing degree of sophistication of user interaction. These are: standard zoom (1), content zoom (2) and local displacement (3) (Figure 4.1).

The general flow shown in Figure 4.1 is as follows:

1. **Standard zoom** A user would typically first zoom to the desired location using the selected extent as spatial reference. This step, called 'standard zoom', denotes the traditional generalisation operations for the current scale at which the map is being represented. Generalisation operations such as selection, simplification and aggregation are mainly applied.
2. **Content zoom** In the second step the user adjusts the generalised content of the map – resulting from the standard zoom in step 1 – to his/her preferences by selecting the content zoom level. 'Content zooming' enables the user to override the result of standard zooming, and add more (as in Figure 4.1) or less foreground objects than recommended by the previous generalisation operations. This would be equivalent to 'content zooming-in' (more foreground objects displayed) and 'content zooming-out' (less foreground objects), respectively. In the user interface of the prototype application the levels for the content zoom and standard zoom, respectively, are adjusted and visually represented by a slider element (Figure 4.4).
3. **Local displacement** 'Content zooming-in' typically displays map content generalised for a higher (i.e. more detailed) LOD with more content resulting in cartographic conflicts that can be resolved by local displacement operations. As a consequence of overlaps resulting from content zooming, the user may request those overlaps to be resolved. Alternatively, overlap resolution is automatically triggered by the system in response to user interaction.

In addition to the above three steps, the methodology also distinguishes between three strategies (columns in Figure 4.2) that differ in the degree of sophistication of the generalisation approach applied, and thus in how they impact on content zooming and exploration. 'Degree of sophistication' gives a qualitative impression of the complexity of generalisation methods applied, from very basic, pre-defined scale ranges to complex and computationally more intensive generalisation operators. The strategies used in Figure 4.2 illustrate how semantic and spatial generalisation are handled in combination with content zooming. The division into three strategies does not exclude any combination between them, which is of course possible.

Strategy A Is based on predefined scale ranges over which particular feature classes are displayed, based on visibility rules (such as in the Mapnik renderer of OSM). As an example of a visibility rule, on maps showing landmarks represented by POIs, feature classes of higher importance – and typically of less frequent occurrence – such as hospitals or train stations, are retained in the lower LOD, as opposed

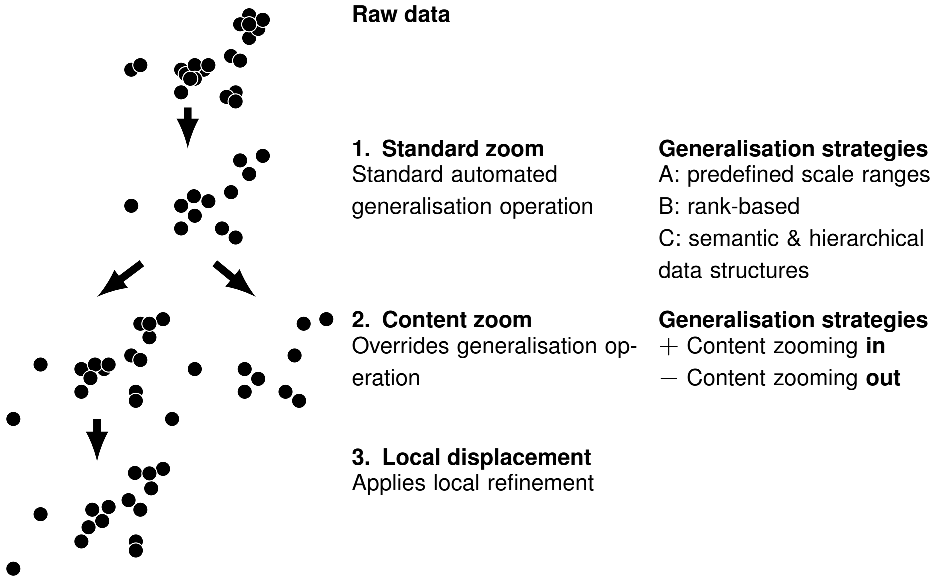


Figure 4.1.: Schematic flow chart for information exploration by content zooming applied to foreground data.

to feature classes of lower importance (as defined in the rule), such as POIs for benches or parking spaces.

Strategy B Denotes generalisation operations (selection) based on ranking, making use of attribute values such as relevance or similarity measures associated with point features. In the case of POIs representing restaurants this might be a popularity rank; in the case of animal observation data this might denote the number of observed animals per observation location.

Strategy C Denotes semantic and spatial hierarchies, based on aggregation or typification operators. In Strategy C points are typically aggregated and typified according to geometric or semantic criteria by semantic and/or spatial distance, applying more complex generalisation operators.

In the following, the three strategies are presented and their application in information exploration is illustrated in combination with the three types of zoom operations.

4.2.1. Strategy A: Generalisation with predefined LODs

A1. Many GIS and mapping applications support the representation of POIs dependent on scale ranges defined in visibility rules. With the standard zoom the POIs are shown only at large scales, while they are hidden at smaller scales to prevent cluttering. The definition of scale ranges of visibility per feature class has been proposed by Cecconi et al. (2002), and Brewer and Buttenfield (2007).

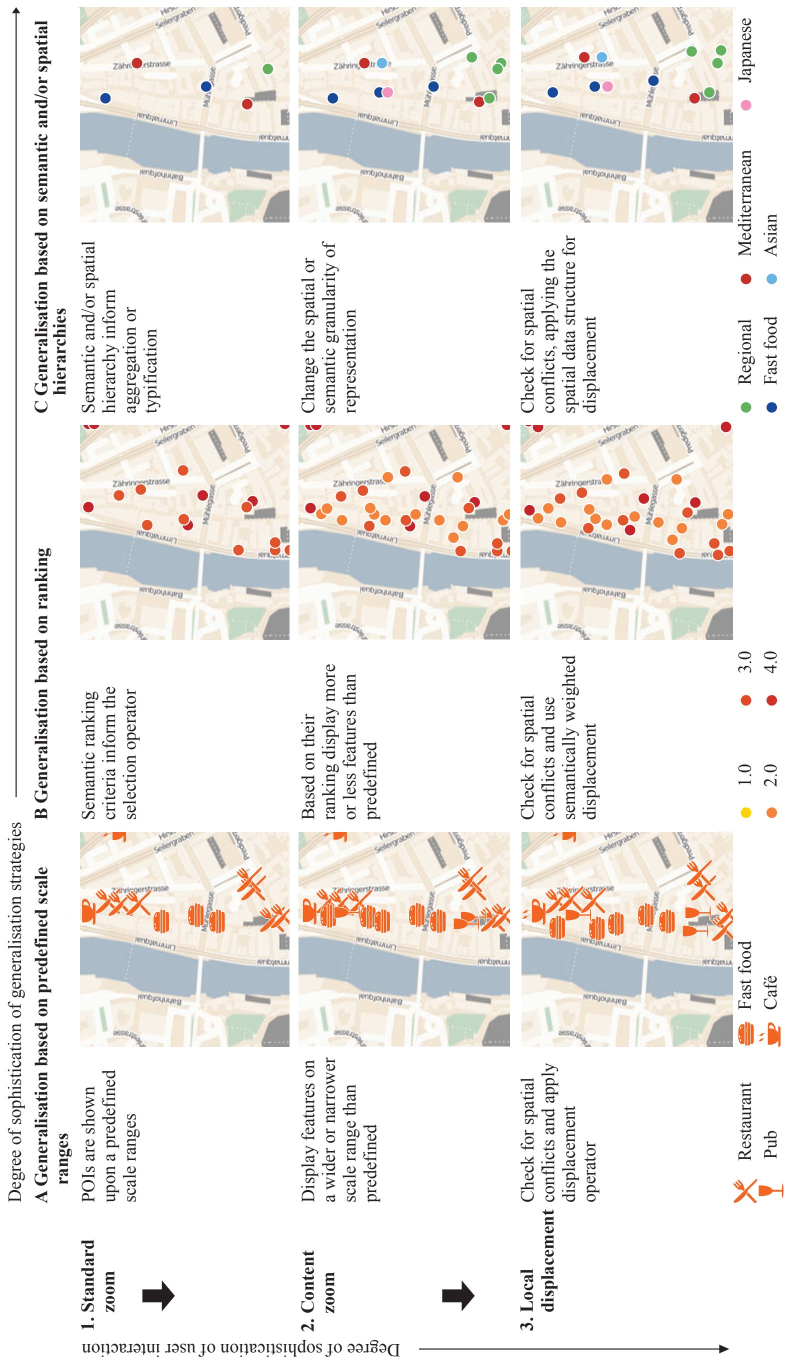


Figure 4.2.: Overview of generalisation approach for information exploration and content zooming

A2. Content zooming gives the user the flexibility to extend the scale range where POIs are shown. Hence, the number of POIs may be increased, effectively widening the scale range at which a particular class of POIs is shown. Conversely, the number of POIs displayed may also be reduced, narrowing the valid scale band (see Section 4.3.4).

A3. If the number of foreground objects is reduced in content zooming, then most probably the legibility constraints are satisfied. Conversely, if content zooming is used to increase the amount of information, POIs are likely to overlap. To remove overlaps, local displacement can be applied automatically such as the radial displacement algorithm by Mackaness and Purves (2001), or the quadtree-based algorithm described in Chapter 6. This type of conflict resolution is applicable only for relatively small overlaps. Alternatively, other generalisation operations have to be applied as well (see Strategy B and C).

4.2.2. Strategy B: Generalisation based on ranking

B1. This strategy considers the generalisation of point features based on semantic ranking criteria such as relevance or similarity measures (Reichenbacher, 2004; De Sabbata and Reichenbacher, 2012). The amount of information represented can be made dependent on a threshold or an interval for the ranking (e.g. show POIs with 80% relevance), or dependent on a fixed number based on the screen size and resolution (e.g. show the 20 most relevant POIs), or based on some selection function, such as the 'Radical Law' of generalisation (Töpfer and Pillewizer, 1966).

Three types of ranking are defined. First, *static semantic ranking* uses fixed ranking orders which may stem from pre-computed measures of similarity between different point features; ranking based on user communities; or ranking based on simple attribute values such as price. Second, there is *dynamic semantic ranking*, where the ranking order may change depending on the user and the context. Third, *spatial ranking* considers spatial dependencies and spatial patterns (e.g. co-location of objects, such as a cash machine in the vicinity of a restaurant and close to a public transport station). To visualise the ranking of POIs a colour ramp or changes of saturation work well, displaying the 'most important' as 'most salient' with respect to colour and contrast (Crease and Reichenbacher, 2011).

B2. The application of content zooming will override these thresholds and allow for a more or less dense portrayal of POIs per LOD. Thus, content zooming provides a simple analysis tool for the identification of POIs according to their ranking (see Section 4.3.5).

B3. Spatial conflicts that might occur as a consequence of the previously applied step can be resolved by a semantically weighted displacement. Higher ranked POIs retain their position, while lower ranked POIs get displaced. If in the content zoom of B2 the number of POIs was increased, this will have caused lower ranked POIs to become visible, making them candidates for displacement.

4.2.3. Strategy C: Generalisation based on semantic and spatial hierarchies

C1. While Strategy A was based on predefined point (sub)sets per LOD, and Strategy B was based on relatively simple selection operations using pre-computed ranking measures, Strategy C introduces aggregation and typification operations, which are more complex. Generalisation based on aggregation or typification allows aggregating multiple POIs, representing them by a place holder POI with a modified position. These operations can be aided by hierarchical spatial data structures such as the quadtree or k-d tree, or semantic data structures, such as the dendrogram. Standard zooming will, by definition, result in a representation without spatial conflicts.

C2. Content zoom may be applied in order to change the spatial granularity of the foreground data in the map (for details on granularity see Section 4.3.6).

C3. In the case of spatial conflicts, displacement can be executed based on the existing hierarchical spatial data structures in a fast and elegant way, exploiting the inherently stored knowledge on proximity (Bereuter and Weibel, 2013).

4.3. Application examples and workflow integration

4.3.1. Development environment and data

The prototype for content zooming was implemented in a development environment using Java and Processing (www.processing.org). An in-depth description of the prototype is provided in the appendix A of this work. The map client, developed for research, features real-time generalisation algorithms Bereuter and Weibel (2011) and an integrated cartographic analysis module Bereuter and Weibel (2013) in addition to the content zooming functionality presented here. For the case studies presented below two different datasets for the foreground data were used (see also Appendix B for a full description of the datasets).

The first dataset originates from OSM and represents POI data for the Canton of Zurich, Switzerland. This POI dataset features several categories, from which a subset of more than 2700 eating places (restaurants, bars, cafés etc.) was extracted as a thematic layer for the presented case study.

The second dataset represents a point collection of observation data about lichens. The point collection used here originates from SwissLichens (Stofer et al., 2012), a database maintaining past and present population distribution of more than 500 different lichen species at over 86000 locations within Switzerland. This dataset features a large set of categories and attributes on each lichen location, as well as information on precision, time and several indicator values. An OSM-based web map tile service was used to provide the background layer.

On the prototype used (Appendix A), for the two datasets the average update time between LODs – including selection, simplification and aggregation – lies within 3 ms for the OSM POI dataset and around 130 ms for the SwissLichens dataset. Including the displacement operator, the update time for the OSM POI dataset is around 13 ms, and 800 ms for SwissLichens data set, as displacement is more costly regarding performance. The initial creation of the quadtree data structure takes approximately 8 ms for the first and 460 ms for the second dataset. More details on the performance is provided in Chapter 8.

Figure 4.3.: Lichen found in the old botanical garden of Zurich. Example of a epiphytic (left) and a saxicolous (right) lichen.

About lichens Lichens are a mini ecosystem of at least two organisms, a fungus (mycobiont) and a photosynthetic partner (photobiont), which is either a green alga or a bacterium capable of photosynthesis (Purvis, 2000).

As a pioneer species lichens are able to grow in extreme climatic conditions. However, due to slow growth and high demand for sunlight, lichens have difficulties in competing with plants, and are therefore often found in ecological niches where plants hardly grow. Even though lichens have no roots, as in the case of plants, the type and the stability of its substrate is a major factor of a lichen habitat. Lichens are found on various types of substrates, such as on trees (epiphytic), on dead wood (lignicole) on stones (saxicole) and on the ground (terricole).

Lichen may be long-lived, but are also vulnerable to environmental disturbances and are therefore often used as indicators in assessing air quality or metal contamination. Lichens respond to environmental changes, their appearance or disappearance indicates changes in their environment and can be used to monitor the health of an environment or ecosystem. Due to their longevity they are a good indicator of the long term ecological stability of their environment (Scheidegger et al., 2002).

4.3.2. User interface

The graphical user interface (GUI) for the content zooming prototype consists of a background map providing the spatial reference, and POI data displayed as point symbols in the foreground. Map user interaction is implemented with a standard pan and zoom interface. The content zoom is represented as a second 'slider' which allows overriding – by adding or removing points – the standard zoom (i.e. the standard generalisation) for the current map extent (Figure 4.6). Changes in map zoom level by using the standard zoom

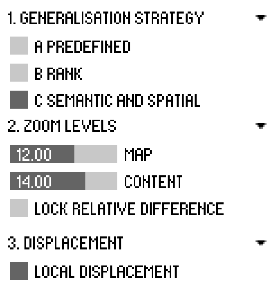


Figure 4.4: User interface to control content zooming in three steps: 1) choice of generalisation strategy, 2) adjustment of zoom levels for map (by standard zoom) and content (by content zoom), 3) optional local displacement. The simplified default view, as shown in Figures 4 to 7, is restricted to step 2 (zoom levels).

reset the content zoom slider to the same (content) zoom level.

In the prototype implementation, the level indicating the standard zoom is the same as map zoom level used in online web map tile services (WMTS) according to the OpenGIS Web Map Tile Service Implementation Standard (Maso et al., 2010). The map zoom level reflects the map scale, where the smallest scale is zoom level 0, representing the whole earth, while the largest zoom level is 20, representing the street level.

Furthermore, the user can apply the same zoom increment to both the map zoom level and content zoom level on an ordinal scale, by selecting the option *lock relative difference*. Thus, for instance, if the relative difference between map zoom level and content zoom level is 2 and the map zoom level is changed from 12 to 13, then the content zoom level will change from 14 to 15. The user can either choose from a range of generalisation algorithms, depending on the theme and purpose of the map, or use the 'workflow view' (shown in Figure 4.4) and select from the three different generalisation strategies offered.

4.3.3. Case studies

The two case studies presented in this section illustrate how content zooming and information exploration works for the three strategies presented in Section 3.

Figure 4.5.: left: LBS case study, tourist in the city of Zurich and possible questions. right: Web case study, researcher with a new spatial data set and possible questions.

The first case study is set within the domain of LBS, envisioning a tourist visiting the City of Zurich, unfamiliar to him, looking for a place to get something to eat. The second case study shows how content zooming can be applied in data exploration within the domain of visual analytics, envisioning a researcher exploring a large spatial dataset in a web environment on a workstation.

4.3.4. Strategy A: Implementing generalisation based on predefined LODs

In Strategy A, POI data of a certain category (e.g. eating places, parking spaces) are shown only up to a predefined LOD. It is the most basic application of content zooming. Content zooming enables the user to over- or underpopulate the predefined point set, to obtain the information he/she requires. If standard zooming (A1) did not include any sort of generalisation, content zooming empowers the user to reduce the amount of information to the desired level. Finally, local displacement helps in both cases to resolve spatial conflicts emerging from the user interaction (A3).

The associated case study pictures a hungry, American tourist named Tom, who is looking for an overview of eating places in downtown Zurich, Switzerland. His mobile mapping application presents, for the selected map extent, a predefined set of categories of eating places (Fig. 4.6a), which leaves him, however, clueless about the actual spatial distribution of eating places. With the content zoom he adjusts the amount of displayed POIs to a higher LOD to see all eating places available in his area, and figures that in the neighbourhood called Niederdorf (around the black dot) his stomach may not be left unsatisfied.

Note the scale bars of the graphical user interface in Figures 4.6a and 4.6b. In Figure 4.6a, both the bar for the map scale and the content zoom scale are equivalent, indicating that the content is generalised and displayed in accordance with the target scale (LOD 14). In Figure 4.6b, however, the content zoom bar is longer, since content zooming now displays the more detailed content of LOD 16.

While the first case study shows content exploration in the information seeking tasks of a mobile user, the second case study focuses on data exploration of a large point collection, where content zooming is applied to the SwissLichens dataset.

Envision a researcher named Ada, exploring a new, unfamiliar dataset for subsequent analysis. The mapping application first presents her with a generalised view off the spatial footprint of the whole dataset (Fig. 4.6c). Content zooming-in gives her a cluttered, yet comprehensive, overview of the overall spatial distribution of lichens (Fig. 4.6d), while content zooming-out (not shown in Fig. 4.6) provides an idea of in which regions of Switzerland the 'hotspots' of lichens are situated, or rather, in which locations – accessible to humans – lichens were observed and/or collected.

4.3.5. Strategy B: Implementing generalisation based on ranking

Content zooming based on ranked point features, be it static or dynamic semantic ranking, and based on attribute data or spatial ranking including spatial dependencies among the point features, supports the user in his/her map reading task by highlighting spatial patterns. The following two applications illustrate how in the domain of LBS and data exploration, Strategy B is applied, and highlights the potential for the use of content zooming.

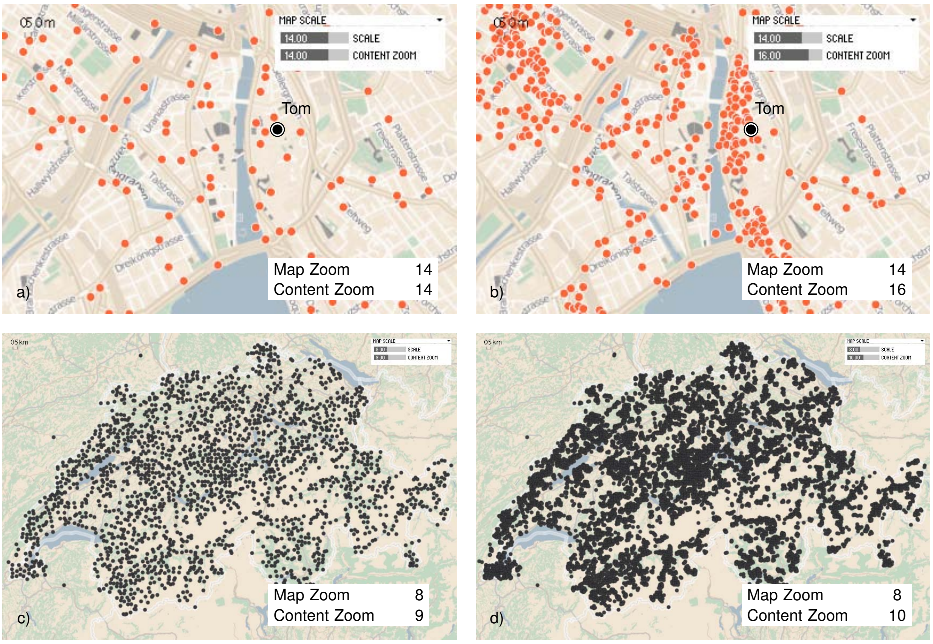


Figure 4.6.: a,b). Eating places in Zurich with (a) and without (b) generalisation of POIs. The black dot denotes Tom's position. c) SwissLichens generalised view d) SwissLichens content zooming to content LOD 10 (instead of 8).

In our first case study Tom got overwhelmed by the vast amount of eating places returned by his mobile application and thus requests a ranked set of eating places that are recommended by some relevance ranking or recommender system. Considering Tom's position and his request, the system applies dynamic semantic ranking and normalises the ranking results to Tom's local neighbourhood. The result for his request (Figure 4.7b) is a ranked, generalised map commensurate with the selected zoom level 15, where the rank is shown in graduated colours.

Tom wants to know where clusters of good eating places are located in the vicinity of each other. He thus applies content zooming to show more results than originally suggested (Figure 4.7a), moving to content zoom level 17, but leaving the map zoom level at 15. Conversely, if he only wanted to see some of the best suggestions he simply reduces the number of POI displayed (Figure 4.7c), by choosing content zoom level 12.

In our second case study, where visual exploration of a large (lichens) data set for research purposes forms the focus, the workflow of Strategy B is similar to the first case study. After getting an overview on the overall data distribution, ranking lichen observations by some attribute – such as a special plant characteristic or the red list status[1]

[1]The 2002 red list of threatened and rare epiphytic and terricolous lichens was drawn up according to the IUCN criteria 2001 (Scheidegger et al., 2002).The International Union for Conservation of Nature (IUCN)

(Scheidegger et al., 2002) – provides a deeper insight into the inspected data collection. And if static semantic ranking is applied the colour scheme used to visualise the ranking attribute remains during the different map interactions.

After Ada gets an impression of the spatial distribution of the lichens in Switzerland, she is especially interested in where hotspots of endangered lichen species, which may need further protection, are located in the Northern part of Switzerland. She therefore applies a ranking based on the red list status of each lichen to get an overview of the distribution of rare and endangered lichens. Not satisfied with the standard view (Figure 4.7e) – showing also less endangered lichen locations – she reduces the content zoom, such that hotspot regions are not further cluttered with locations of less endangered lichens (Figure 4.7d). In a region she knows close to Lucerne she expects to find more lichens. she then realizes, by content zooming out, that there is a larger number of lichen observations but only few are endangered (Figure 4.7f). This way, she can flexibly explore the dataset, not only spatially, but also with respect to content.

As we can see from these two examples, content zooming maintains the spatial frame of reference – by retaining the map zoom level – but provides the user with the capability to fine-tune the ranked generalisation results to his/her needs.

4.3.6. Strategy C: Implementing generalisation based on semantic and spatial hierarchies

In Strategy C a distinction is made between two types of hierarchical generalisation, relating to semantic granularity and spatial granularity (i.e. density), respectively, and thus two scenarios are presented. Semantic granularity denotes the degree of classification detail of categorical data; for instance, representing a coarse-grained classification of 'eating places' vs. a fine-grained classification of cafés, pubs, fast food, restaurants etc., or different types of cuisine. Conversely, spatial granularity denotes the density of elements displayed and varied by generalisation operators.

Here semantic granularity is exemplified by the LBS case study, while spatial granularity is demonstrated by the case study on data exploration. While Tom figured out that all highly recommended eating places are not nearby, he got really terribly hungry, and cannot wait any longer. The next map he tries shows possible locations (Figure 4.8a, Strategy C1), but not of what type of eating place they are. Content zooming allows him to increase the semantic granularity (Figure 4.8b, Strategy C2) and provides him with the necessary information to select a place nearby where he can grab some fast food instead of haute cuisine.

Depending on the generalisation parameters applied and constraint settings employed (alternatively set by the user or the system), the foreground point features may have been spatially aggregated or typified such as in Figure 4.9 (left, Strategy C1), based on semantic hierarchies or based on density and feature co-location represented in hierarchical spatial data structures such as quadtrees or k-d trees (Cecconi, 2003; Bereuter and Weibel, 2011). In such a case, applying content zooming will provide the user with the capacity to comprehend the underlying generalisation process (Figure 4.9b). Furthermore, gen-

publishes the IUCN red list of threatened species and tries to assess the conservation status of species.

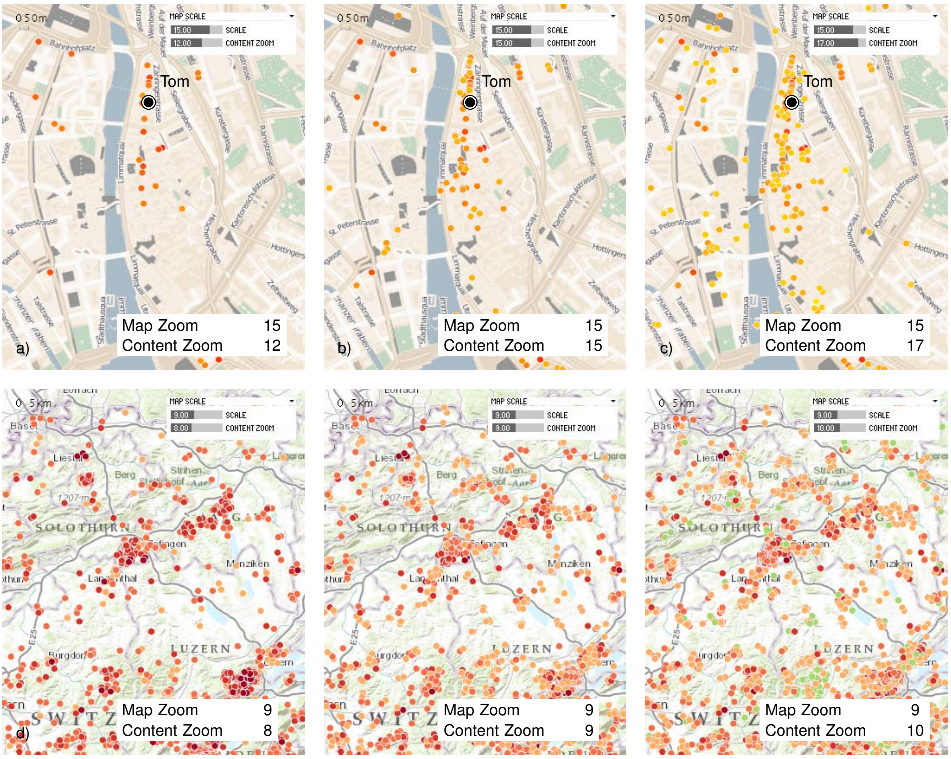


Figure 4.7.: a, d) 'underpopulated', b, e) standard zoom, c, f) 'overpopulated' content zoom applied to POI data (top row) and point collection (bottom row), respectively. Red colour reflects the highest rank for both use cases.

eralisation algorithms based on spatial distribution, such as aggregation using quadtrees, tend to retain more point features (see Chapter 6). So, the user may want to apply content zooming to reduce that load, effectively reducing the spatial granularity, while maintaining the spatial distribution.

Back to Ada, she just applied a selection operator to the SwissLichens dataset, using a quadtree-based algorithm. This algorithm takes into account spatial properties, by only removing points if cartographic constraints – minimum distance criteria – are not satisfied. Thus, the spatial distribution in the local neighbourhood is maintained, giving a better impression of the spatial coverage than by applying a ranking scheme globally. Ada furthermore sets the parameters of the selection operator to favour and retain locations of endangered lichens. By content zooming in and out, she balances the spatial granularity of the point distribution. She may even use other generalisation operators to further investigate the dataset, such as displacement, to retain as many lichen locations as possible; or aggregation, by aggregating close points to form graduated map symbols, thus

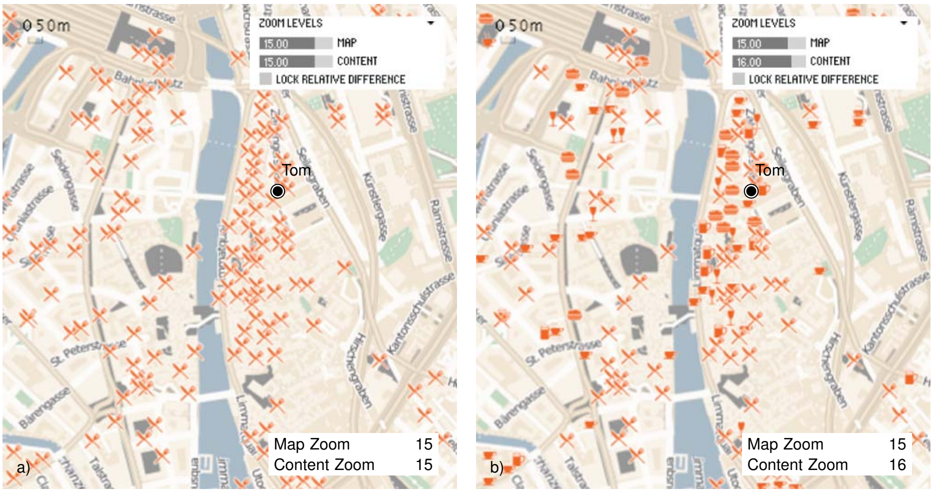


Figure 4.8.: Content zooming changes semantic granularity of POIs. Lower (a) vs. higher (b) semantic granularity

highlighting hotspots.

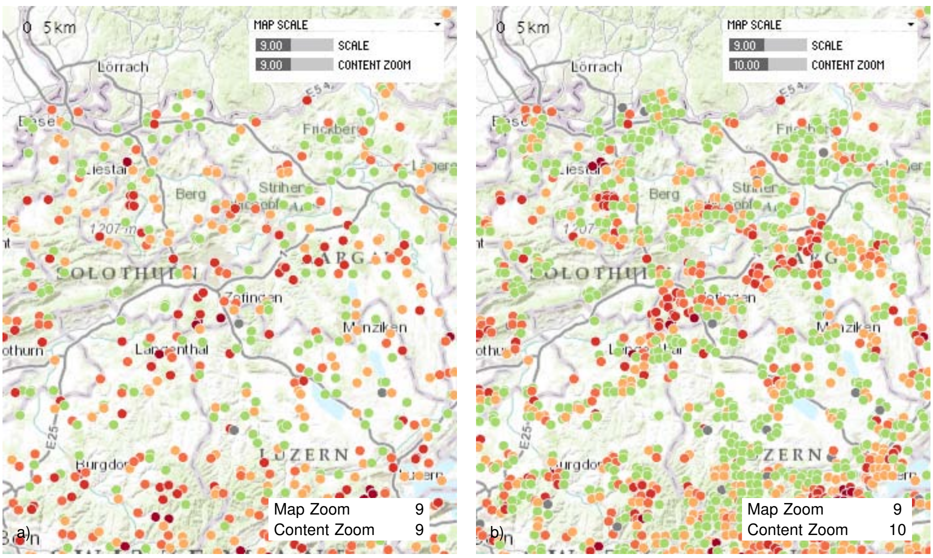


Figure 4.9.: Content zooming changes the spatial granularity of lichen observations. a) Standard spatial granularity. b) Higher spatial granularity.

4.3.7. A content zooming workflow

The first use case study shown above, illustrated for the domain of LBS how spatial decision making is supported by analysing spatial relationships and phenomena. The second case study demonstrated, for the domain of visual analytics, how web-based data exploration is simplified. Both case studies illustrated how map interaction can benefit from a looser connection between spatial map scale vs. LOD of the thematic foreground data. Nevertheless, with the standard zoom operation, which integrates generalisation operations of various degrees of sophistication and computational demand, the link to map scale is maintained. Thus, the case studies suggest that the various operations could be integrated to a 'workflow' of content zooming and information exploration, as will be shown below.

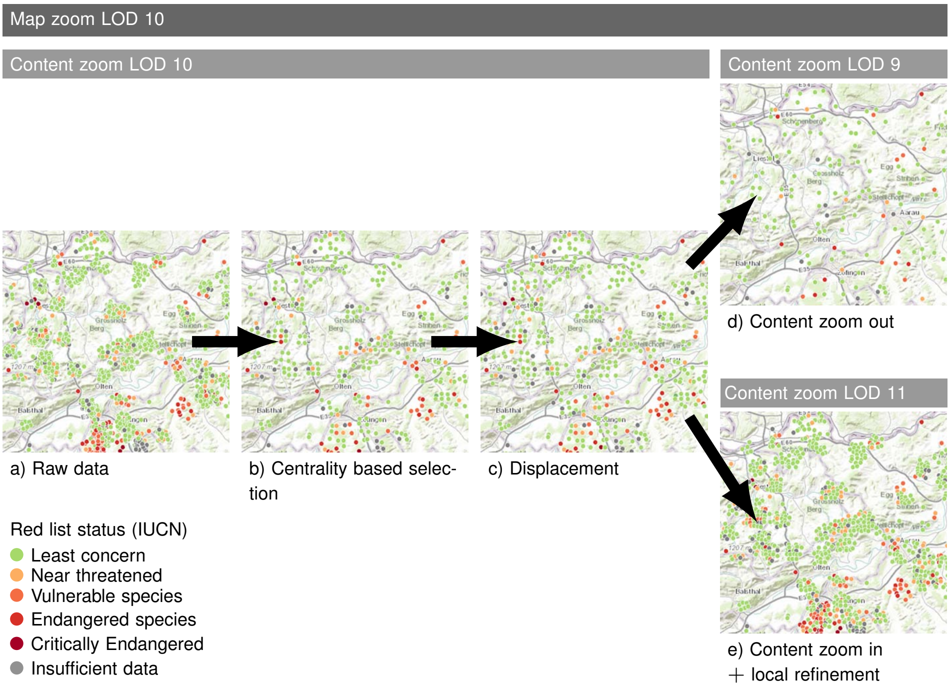


Figure 4.10.: Sample content zooming workflow.

In Figure 4.10 a series of steps illustrates how different generalisation operations may be applied and chained together, applied to the SwissLichens dataset with the aim of highlighting regions where endangered lichen species are located. Based on this example the overall workflow is shown. However, Figure 4.10 does not show any backtracking and iterations, which would of course be possible at any point of the workflow, to change generalisation parameters, the map zoom level, content zoom level, and change or chain further generalisation operators. In the first step of the workflow (Fig. 4.10b) centrality-based simplification (see Chapter 6) is used to relax spatial conflicts, whilst largely maintaining the spatial distribution of the input data. Different generalisation strategies (see Section 4.2) and extensions, such as co-location rules, may be applied using generalisation operators that reduce the amount of data. Remaining spatial conflicts can then be further resolved by the displacement algorithm (Fig. 4.10c; Chapter 6). Figures 4.10d,e show the next steps of the workflow, where zooming in and out on the content, respectively, enables the user to override the 'standard' settings of the applied map generalisation operator without changing the map scale. Note that content zooming can follow any generalisation operator, at any stage of the content zooming workflow. Zooming in on the content (Fig. 4.10e) increases spatial granularity but decreases map legibility. In order to alleviate spatial conflicts, several options can be applied: local displacement to reduce the degree of overlap between point symbols; a reduction of the map symbol size; a change in symbol type; or the presentation of the foreground map can be substituted by a kernel density surface. The latter option may be especially useful if a user zooms in on the content over several levels, which is likely to result in a highly cluttered map of point symbols.

4.4. Analysis and discussion

Based on the case studies presented above, several aspects of the proposed methodology and the applied generalisation algorithms were analysed, such as the amount of data shown per map scale (or zoom levels), counts of spatial conflicts, and their resolution, as well as the density distribution and the changes to the density distribution between map scales.

The reduction of data over the range of scales (or zoom levels) for the SwissLichens dataset is illustrated in Figure 9 for a group of generalisation algorithms belonging to Strategy C (cf. Section 4.2). It shows the difference in the number of points retained for different quadtree-based generalisation algorithms (see Chapter 6), and the effects of content zooming in and out. Figure 4.11a shows nicely how the two blue curves of the standard zoom take a middle position, while content zooming shifts the curves to the left and right, respectively. Zooming in on the content (green lines) shifts the curves to the left, indicating a higher number of points retained and thus 'overpopulation'. Zooming out on the content (pink lines) shifts the curves to the right, meaning 'underpopulation' of the map with points. In each pair of curves, the curve for displacement lies to the left of the curve for the corresponding point reduction algorithms (selection, simplification, aggregation). Displacement increases the holding capacity of the map, due to the resolution of overlaps between point symbols, which makes room for more point symbols to be displayed. For comparison, Figure 4.11a also shows the data reduction curve of the Radical Law (Töpfer and Pillewizer, 1966). For a discussion and analysis of the data reduction behaviour and quadtree-based generalisation see Section 8.4.

For cartographic conflicts (i.e. symbol overlaps) a similar behaviour can be observed for the different zoom levels, as illustrated for the case of a quadtree-based selection operator (Fig. 4.11b). The basic algorithm does not entirely remove all symbol overlaps by default, due to the fact that it retains one point location per quad node up to a tree depth defined depending on the current zoom level. It does not consider *per se* potential

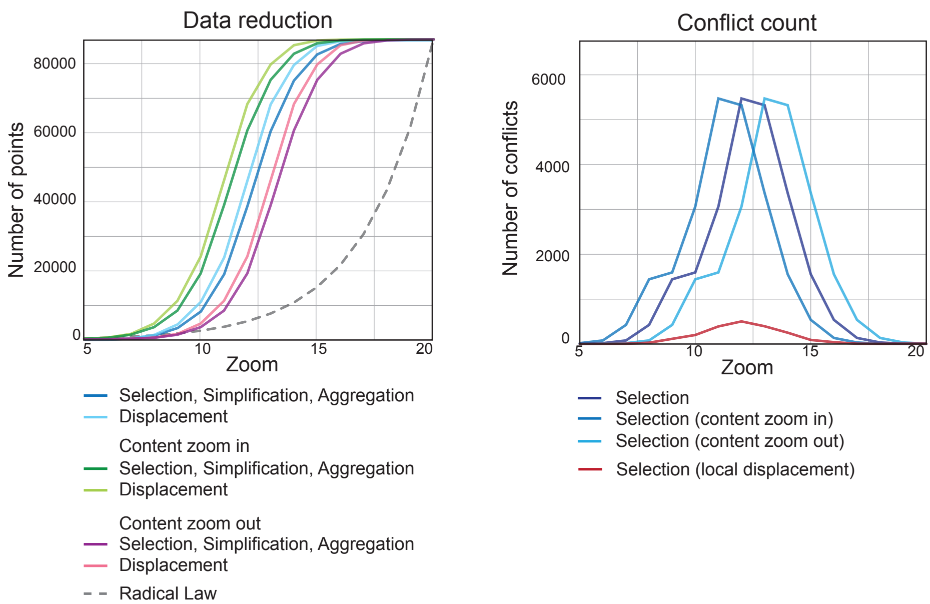


Figure 4.11.: . a) Data reduction per zoom level, for quadtree-based algorithms for point reduction (selection, simplification, aggregation) and displacement: standard zooming (blue lines), content zooming in by one zoom level (green lines), content zooming out by one zoom level (pink lines). For comparison, the dashed line shows the data reduction curve of the Radical Law. b) Conflict count per zoom level for standard zoom and content zooming in/out (blue lines), and standard zoom with conflict constraints (red line). Both graphs are based on the SwissLichens dataset.

symbol overlaps from retained points contained in neighbouring quadnodes (see Chapter 6). This can, for instance, be alleviated by a local displacement algorithm that checks for collisions in neighbouring nodes. Figure 4.11b shows the evolution and reduction of cartographic conflicts over different zoom levels, for standard zooming ('Selection') and for content zooming (in/out). As the red curve for local displacement demonstrates for 'normal' selection, this additional generalisation operation has a significant effect.

In Figure 10 the red dots highlight overlaps of map symbols, arising in different zooming operations. Figure 4.12b shows the situation for standard zooming to level 10, after generalisation that merely uses a selection operator. Since no displacement operator was applied, some symbol overlaps remain. Once local displacement has been applied, these overlaps disappear (Fig. 4.12c). Conversely, if one zooms in on the content to level 11 (Fig. 4.12a), nearly all map symbols overlap for this clustered point distribution. A closer look to the degree of overlap – ranging from symbols that merely touch to nearly full overlap – shows that symbols overlap mostly happens only to a small degree, and still show relatively good visual separation, thanks to the underlying generalisation operator (quadtree-based selection).

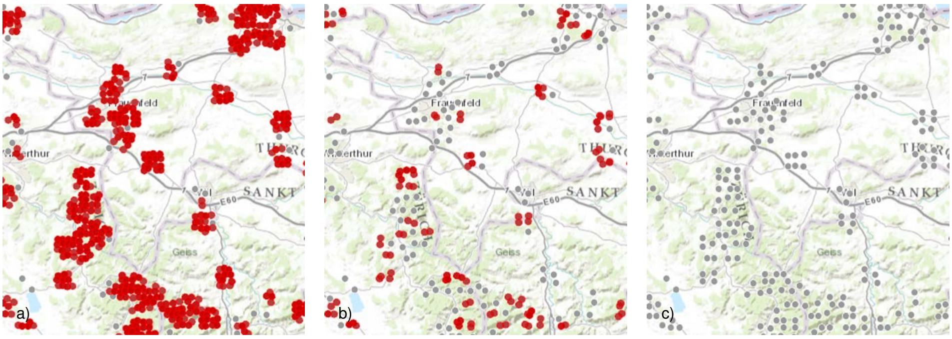


Figure 4.12.: Figure 10: SwissLichens map zoom level 10, generalisation Strategy C, quadtree-based selection operator. Red dots indicate cartographic conflicts. a) Content zoom level 11, b) standard zoom (level 10), c) standard zoom level 10 with local displacement.

4.5. Concluding remarks

The mission of generalisation, in its classical form, is to visualise map content at different levels of detail, commensurate with the target scale and map purpose. This conventional approach, however, unnecessarily restricts the mobile or web map user in his/her abilities to interact with a map. Map users are very likely operating in an exploratory information-seeking mode, and are nowadays used to highly interactive interfaces. Furthermore, good cartographic generalisation is hard to achieve automatically and even more so in real-time, which especially holds if map content needs to be adaptable as a consequence of user interaction.

Thus, this chapter started off from a radically different perspective on map generalisation: to give the user the choice of changing the amount and granularity of foreground information presented, independently from the geometric map scale. This is in contrast to classical map generalisation where a change in map scale always affects the amount and granularity of map content and cannot be manipulated in a straight-forward way by the map user. The assumption underlying this approach is that unconstrained exploration of map content based on user interaction may cartographically not provide the user with an entirely attractive and readable map solution, but in combination with real-time generalisation methods, it will provide the user with more adaptable and exploitable map content.

In response to the above assumption, this chapter proposed a novel methodology for information exploration in web and mobile maps, integrating several generalisation and zooming operations in order to better support mobile users in their information seeking and data exploration tasks. In particular, the technique of decoupling content zooming from spatial zooming was introduced, as a means of providing the user with the ability to adapt map generalisation to his/her needs.

Content related zooming thus enables adjustment of the amount and the granularity (see Section 4.3.6) of the foreground content (point features, in this case) and personalisation of the content to the task at hand. By providing the capability of flexibly increasing or reducing the spatial density and semantic granularity of a map, the relevance of the content can be better explored than is the case with automated generalisation alone. Hence, the generalisation functionalities of a mobile or a desktop system are complemented and extended.

Chapter 5.

Analytical toolbox

The question "‘How to choose suitable algorithms for real-time generalisation of point data?"’ leads to the investigation of the cartographic performance of the generalisation algorithms.

Cartographic map generalisation is an inherently subjective process, hence its evaluation is also ultimately subjective. But in order to compare generalisation output and to evaluate the available algorithms for point data generalisation, methods and measures are required for the comparison of algorithms, in order to spot the strength and weaknesses of the algorithms and propose extensions or alternate methods. This chapter presents a workbench system of diagnostic tools that can be used to quantitatively and visually assess point generalisation algorithms. The purpose of the diagnostic tools is not so much conflict detection (e.g. overlap of point symbols) in actual generalisation situations, but to comparatively characterise the behaviour of point generalisation algorithms relative to certain desirable properties, such as maintenance of the distribution and density of the input data.

In the remainder of this chapter, an introduction of relevant cartographic analysis is given (Section 5.1), followed by the introduction of the tools composing the diagnostic toolbox (Section 5.2), and concluding with a small demonstrative set of examples to illustrate the application of the tools in assessing the characteristics of point generalisation algorithms (Section5.3).

5.1. Background: Cartographic analysis

5.1.1. Evaluation strategies

A possible answer to the subjectivity of the evaluation of cartographic generalisation is to resort exclusively to visual judgement by expert cartographers. However, while expert judgement is certainly valuable and necessary to get the ’full picture’, it also implies serious problems in comparing alternative generalisation solutions, as the criteria used by different experts are not clear. Hence, some researchers have proposed strategies to render the evaluation of generalisation solutions more objective and comparable. Stoter et al. (2009) have brought objectivity into their evaluation procedure by using a set of constraints that were defined and assessed by cartographic experts. Bard (2004a) proposed and implemented a detailed evaluation procedure that characterises the state of map objects before and after generalisation, and compares these states against a set of evaluation functions that define how the generalisation result should look like according to cartographic rules.

The objective, however, is different from the above studies. It lies not in evaluating how well specific cartographic conflicts have been detected and resolved, but in knowing how a particular algorithm performs in a more general way; that is, what its characteristics, strengths and weaknesses are when generalising a set of points across a range of scales. This should allow the comparison of different algorithms and their behaviour.

5.1.2. Measures and tools for point data

Following the objective outlined in the previous section, any method is possible that allows the study of desirable properties of point data generalisation. Thus dealing with point sets, any method from point pattern analysis (O'Sullivan and Unwin, 2010) is potentially useful. For the analysis of point generalisation algorithms of particular interest are so-called first-order effects – patterns within the point distribution – rather than second-order effects, which are express interactions with other variables such as background data (e.g. street network, or land-use polygons) (O'Sullivan and Unwin, 2010). A wide variety of measures for point pattern analysis have been developed over the past few decades, and are documented in textbooks such as Ebdon (1985) or O'Sullivan and Unwin (2010). Some measures have been developed more specifically in the cartographic literature, for instance entropy-based measures that express the spatial distribution of points (Li and Huang, 2002; Stigmar and Harrie, 2011). Voronoi polygons give an immediate impression of neighbourhoods but also of the point density, and have thus been used in several studies (Li and Huang, 2002,e.g.). Graph measures provide hints on the internal structure of the point set (Wasserman and Faust, 1994). They are, however, heavily influenced by the use of the Delaunay triangulation of the input points that is used to build the graph. Their value therefore is limited. The Delaunay triangulation can, however, also be used as an auxiliary data structure for other purposes, such as the generation of distance maps between neighbouring points. And, finally, point densities can also be translated into density maps or surfaces by means of kernel density estimation (O'Sullivan and Unwin, 2010).

5.2. Diagnostic toolbox

With the aim of assessing the properties of the generalisation algorithms presented in Chapter 2, using the evaluation strategy outlined in Section 5.1.1, we developed a toolbox containing diagnostic measures and tools of the types reviewed in Section 5.1.2. The toolbox was implemented partly in Java and R, an open source statistics system offering rich libraries for spatial statistics (Bivand et al., 2008). In the following, an overview of some of the main measures and visualisation techniques that have been implemented is given. The presented measures are subdivided into two broad classes: Global measures and local measures.

5.2.1. Global measures

Global measures yield a single number that attempts to summarise a particular geometric property for the entire map (Table 5.1). These measures are particularly helpful when studied in evolution across a range of scales, or in comparison across different generalisation algorithms, or different input data sets with different distribution characteristics (cf. Figure 5.1). For instance, comparing the data reduction rate in relation to zoom level over a set of algorithms, or how many overlaps between point symbols are being resolved. In Table 5.1, each of the measures is assigned to a category and corresponding literature that relates to the geometric property that should be captured by the particular measure. Global measures give a good first estimate and summary of an aspect of a generalisation algorithm. However, the reduction to one single measure in effect misses out on local fine-grained information, that is often necessary for the cartometric analysis.

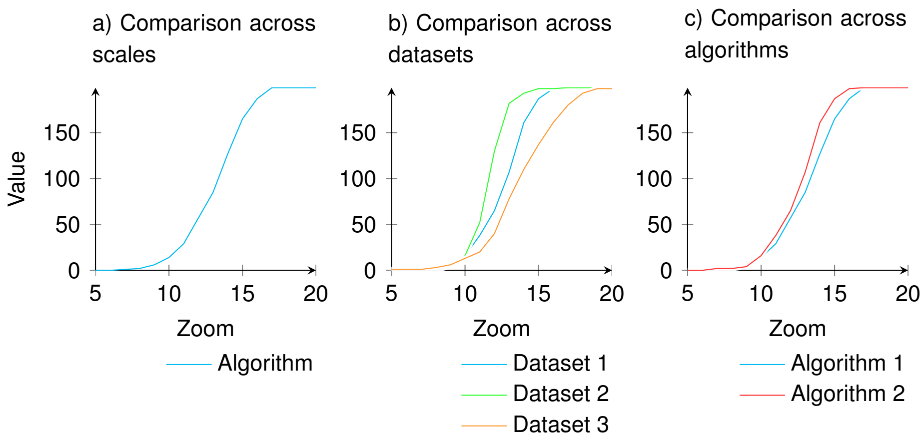


Figure 5.1.: Example of comparison of global measures across scales per algorithm, or dataset.

5.2.2. Local measures

Local measures are valid in a local context. They are listed in Table 5.1. Kernel density estimation (KDE) yields a field of density values at any location across the map, typically approximated by a raster. In the case of proximity maps, the values represent the distance to the closest point at any location in the map (thus bearing similarity with a raster version of the Voronoi diagram). Both density maps and proximity maps can also be subtracted from each other to generate difference maps, and thus identify where change happened between different scales or algorithms. Since local measures can no longer be expressed by a single number, they lend themselves naturally to visualisation, as will be shown in selected examples in the next section.

Besides density and proximity maps, several measures exist in the spatial statistics literature that have been reviewed by Boots and Okabe (2007) and summarised under the heading of Local Spatial Statistical Analysis (LoSSA). These measures can either be computed on the point set or on auxiliary data structures derived thereof, such as the Voronoi diagram. Examples include the local *Moran's I* and local *Geary's C* for spatial association, and local clustering measures and algorithms (Ankerst et al., 1999; Ester

Table 5.1.: Global and local measures for characterising point set distributions in the diagnostic toolbox. Including details of the programming language used.

Categories	**Measures**	**References**	**Code** Java	R
Global measures				
Amount of information	No. of elements		x	x
Centrality	Mean, median, mode center	Ebdon (1985)	x	x
Dispersion	Standard distance	Ebdon (1985)	x	x
	Standard deviational ellipse	Ebdon (1985)	x	
Proximity	No. of nearest Neighbor distances below threshold / degree of overlap	Stigmar and Harrie (2011)	x	
	Nearest neighbor distances distribution		x	x
	Nearest Neighbor Index (NNI)	O'Sullivan and Unwin (2010)	x	x
Spatial distribution	Thematical entropy	Sukhov (1970); Li and Huang (2002); Stigmar and Harrie (2011)	x	x
	Topological entropy	Neumann (1994); Bj (1996); Stigmar and Harrie (2011)	x	x
	Positional entropy	Bj (1996)	x	x
	Geometric entropy	Neumann (1994); Bj (1996); Stigmar and Harrie (2011)	x	x
	Geometric ratio	Li and Huang (2002)	x	x
	Average neighbors	Li and Huang (2002)	x	x
Density	Delaunay triangle Size Distribution		x	x
	Voronoi cell size distribution		x	x
Graph measures	No. of vertices, edges		x	x
	Degree		x	x
	Density			x
	Transitivity	Wasserman and Faust (1994)		x
	Diameter			x
	Shortest path			x
	Minimum spanning tree	Prim (1957)		x
Local measures				
Density	Density Maps (KDE)	O'Sullivan and Unwin (2010)	x	x
	Density differences between scales		x	x
	Density differences between algorithms		x	x
	Local voronoi cell variation		x	x
Proximity	Distance maps			x
	Distance maps differences between scales			x
	Distance maps differences between algorithms			x
	Nearest neighbor link map		x	x
Cluster	K-means, DBScan, Optics	Ankerst et al. (1999); Ester et al. (1996); MacQueen (1967)	x	
LoSSA	Local spatial statistical analysis	Boots and Okabe (2007)		x

et al., 1996; MacQueen, 1967).

Also, additional measures and visualisation techniques would be possible on the basis of those presented here. For instance, geomorphometric statistics could be computed on the density surfaces. Based on experience using the above measures and methods, however, those mentioned above represent a useful set of diagnostic tools to assess the differences and qualities of point data generalisation algorithms.

5.3. Working with the diagnostic toolbox

Following the summary of tools available in the diagnostic toolbox, this section shows a small demonstration of how these tools may be applied in investigating a set of generalisation algorithms.

With a set of multiples over a range of scales, generalisation results can be effectively compared. In classic generalisation literature these multiples typically show the state of a map before and after generalisation, either at the original source scale or at the target scale. Often generalisation results are shown at target and at source scale, to provide the details and compare directly with the source data, and furthermore to provide an impression of the generalisation result at the destination scale. Figure 5.2 shows a set of multiples with a selected set of POI data for the greater area of Zurich at zoom level 13 (ca. 1:72'200, see Appendix B). Generalisation results of two algorithms – centrality-based simplification based on a quadtree index (see Section 6.4.1) structure, and typification based on mesh simplification (Burghardt and Cecconi, 2007) – are compared over three consecutive generalisatoin zoom levels displayed at zoom level 13. The direct visual confrontation highlights details, local differences and the sensitivity of the algorithms to the spatial distribution of the input data.

A generalisation algorithm may show a different behaviour depending on the spatial distribution of the point dataset (strongly clustered, random or uniform distribution). A comparison of generalisation results, based on a set of multiples and different datasets, highlights the sensitivity of the algorithm to the spatial distribution of the input data.

As the possible combinations of all algorithms against all data sets and measures would be prohibitive given the purpose of this section, the following representative of a global measure is restricted to two generalisation algorithms: Typification based on mesh simplification (Burghardt and Cecconi, 2007) and centrality-based simplification based on a quadtree.

It should be noted that in the examples shown in Figure 5.2 and 5.3, the mesh simplification algorithm by Burghardt and Cecconi is applied without the density correction factor that prevents points in dense areas from being excessively eliminated. This is done for didactic purposes, in order to show more vividly the effects of excessive homogenisation and loss of the characteristic spatial distribution of the input data. With the correction factor on, the results would be clearly better, as shown in Burghardt and Cecconi (2007).

In Figures 5.2, it is obvious that the two algorithms have different behaviour in typifying the original point data set. With the application of global measures that are capable of capturing the regularity of a spatial distribution, these differences should become noticeable.

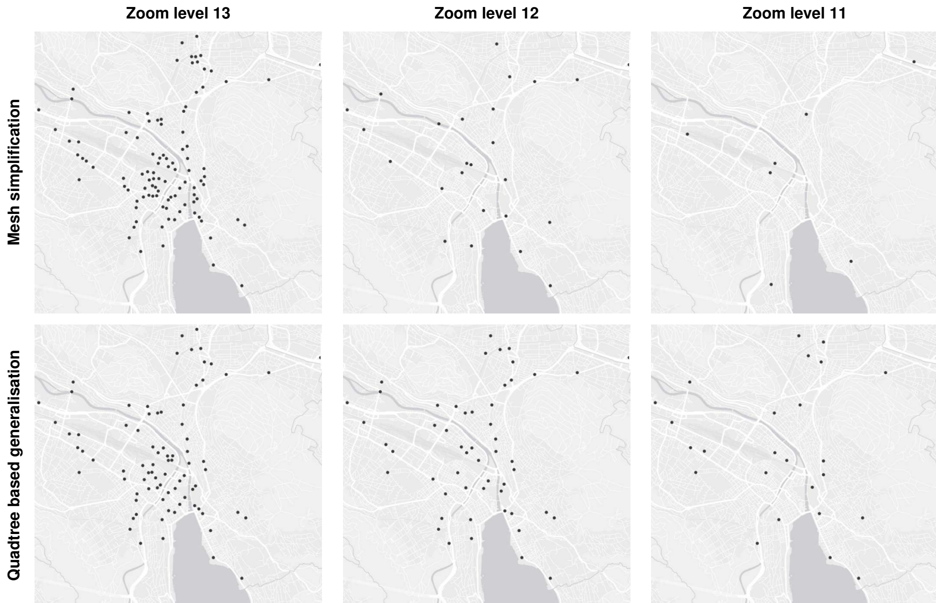


Figure 5.2.: Visual comparison between two algorithms, centrality-based simplification and typification based on mesh simplification (Burghardt and Cecconi, 2007, density correction factor turned off) at the same source scale (zoom level 13), for zoom levels 13,12 and 11.

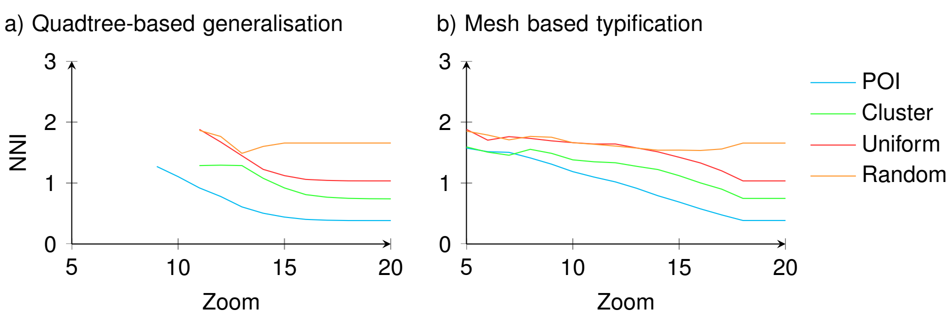


Figure 5.3.: Evolution of the Nearest Neighbour Index (NNI, see Equation 8.5) for the typification based on mesh simplification (a) and centrality-based simplification based on a quadtree (b) for POI data (restaurants) in the city of Zurich, and three artificially generated datasets with different distribution patterns (random, uniform and cluster)

Figure 5.3 shows the evolution across a range of scales for the Nearest Neighbour Index R (Ebdon, 1985), computed on four different point sets generated by the same generalisation algorithm as in the previous figures. The higher the values of R, the more uniform the point distribution that it measures. As the graphs in Figure 5.3 show, the mesh simplification algorithm approaches the uniform point distribution more rapidly and to a greater degree, independently of the input data set used (once again, however, this effect is mainly due to the density correction factor being turned off).

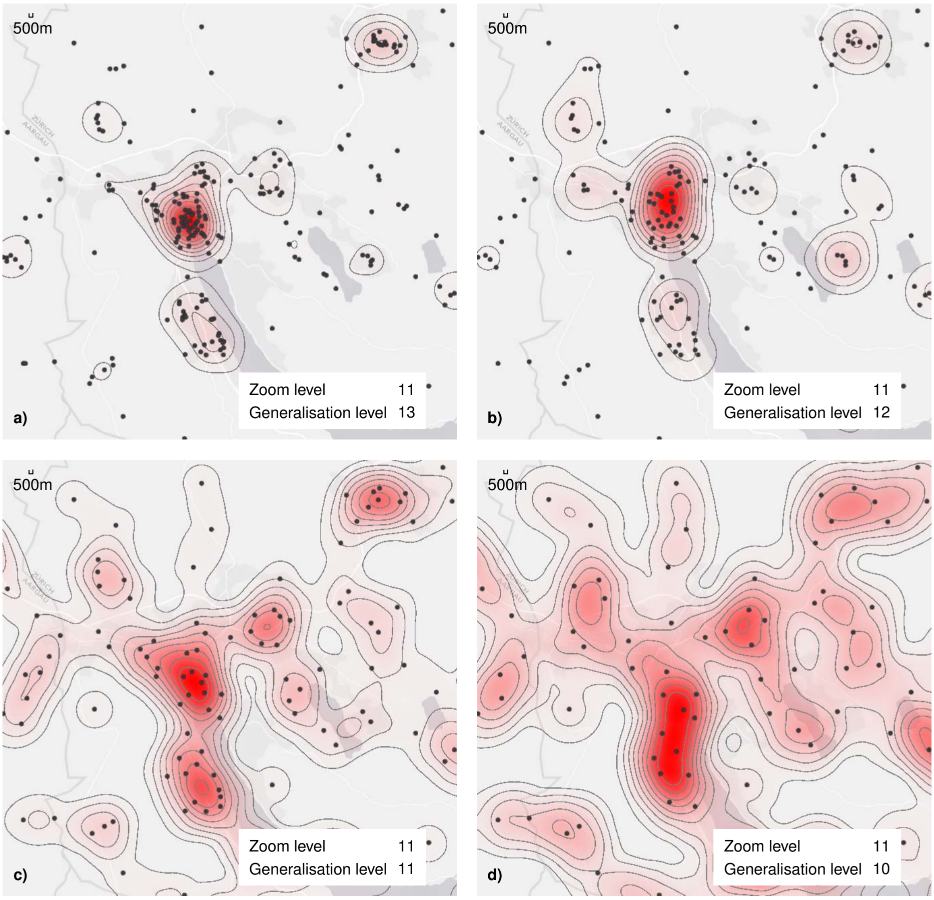


Figure 5.4.: Density maps (KDE) for four scales of points generated by the quadtree aggregation algorithm (Zurich POI data set).

The next few figures demonstrate how local variations in the results of generalisation algorithms can be quantified and visualised. Figure 5.4 shows the density surfaces computed by a kernel density estimation KDE (O'Sullivan and Unwin, 2010) for a portion of the Zurich POI data set at four different scales, generated by the quadtree selection algorithm. Visual comparison of the four sub-figures allows the assessment of where changes took place in the scale transitions.

Difference maps between the density surfaces are a further possibility for assessing

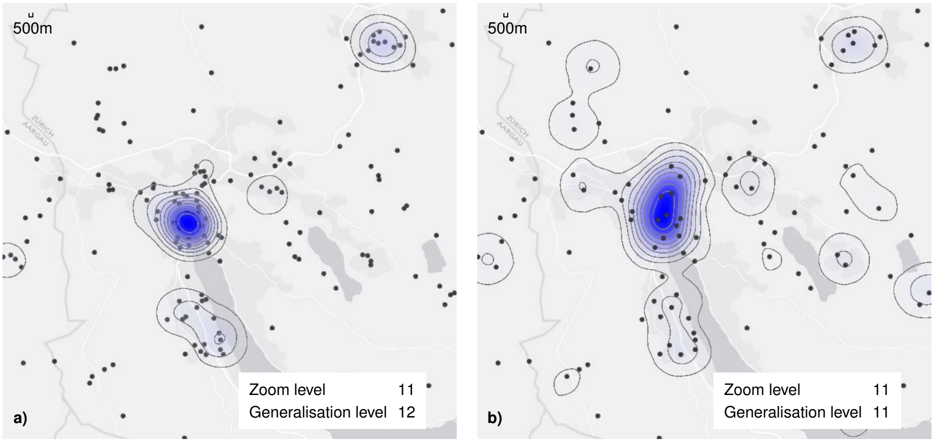


Figure 5.5.: Difference maps between the density surfaces KDE between generalisation level 13 and 12 (a) and 12 and 11 (b) and between Figure 5.4b and c and c and d respectively.

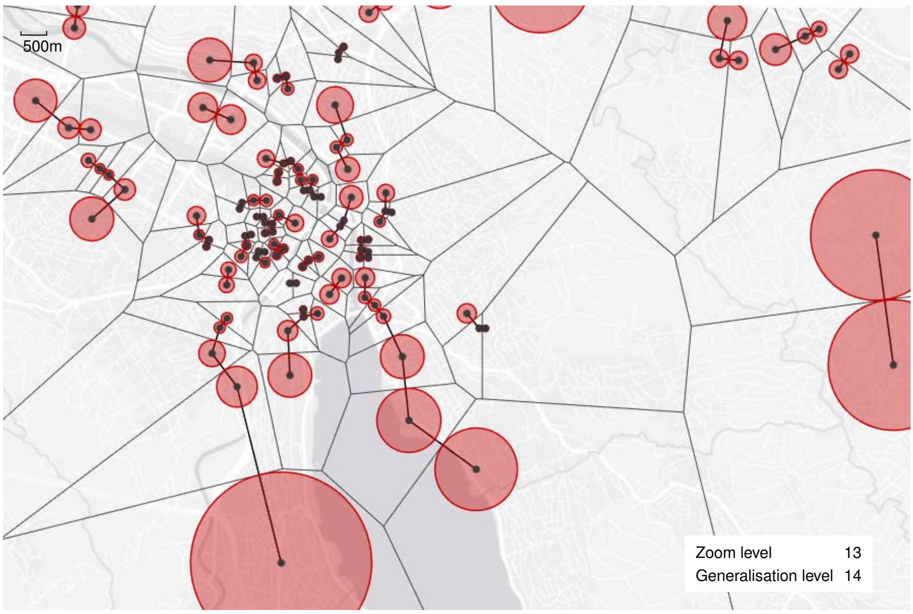


Figure 5.6.: Nearest neighbour link map, highlighting the distance to the nearest neighbour with scaled circles, on top of a Voronoi tesselation.

local changes between different map scales and thus provide a more immediate impression of the local evolution of point densities over a progression of scales with a single algorithm (Figure 5.5). Likewise, difference maps can also be computed between the solutions obtained by different generalisation algorithms for a given scale.

The example of Figure 5.6 illustrates the nearest neighbour link map. The nearest neighbour link map shows links between points that are nearest neighbours, and it further depicts half the distance to the nearest neighbours by circles, to give both a qualitative and a quantitative impression of proximity relations. With the Voronoi tessellation shown in the background, point features violating cartographic constraints in subsequent scales can be identified visually.

Similarly, our last example shows a visual cluster analysis based on the Optics algorithm (Ankerst et al., 1999) a density-based clustering method and an extension of the DBSCAN algorithm by Ester et al. (1996). Clustering provides a good means of assessing how well generalisation operators maintain concentrations present in the dataset.

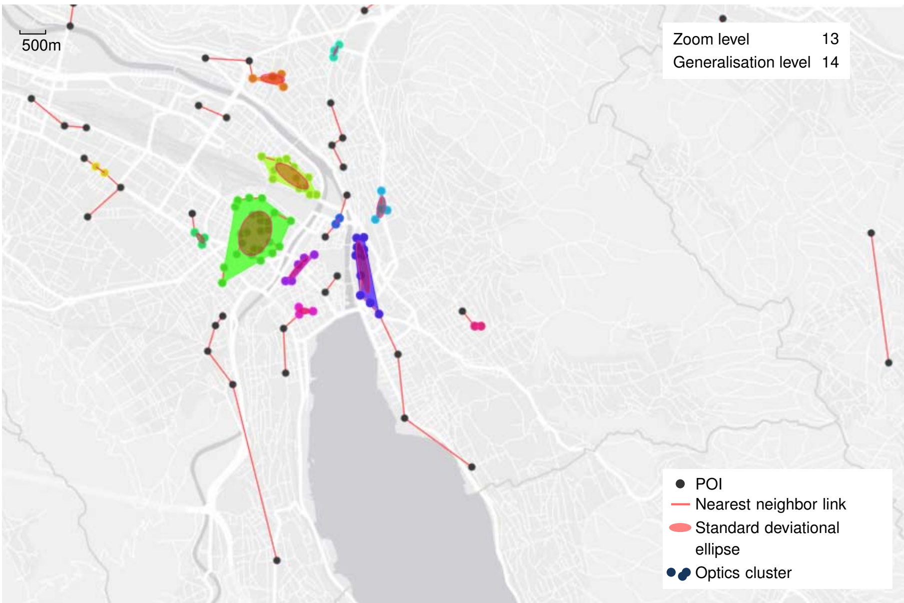


Figure 5.7.: Visual cluster analysis, based on the Optics algorithm, parameterised with ε = 20 pixels and minimum cluster size of 3. For each cluster a standard deviational ellipse is shown and the nearest neighbors of the points are emphasized with a link

It should be noted that while the local measures in the above examples have been solely used for visualisation purposes, they could (and should) equally be used as a basis for further quantitative analysis.

5.4. Concluding remarks

This chapter reviewed the tools for assessing the cartographic quality of real-time algorithms for point generalisation, and the means by which they work. It furthermore presented a prototype system (see Appendix A) offering a set of diagnostic tools that allow comparative characterisation of the behaviour of point generalisation algorithms relative to certain desirable properties, such as the maintenance of the distribution and density of the input data. The system was implemented in Java and R (see Table 5.1). While R is incredibly rich in spatial statistics and spatial analysis techniques (see e.g. Bivand et al., 2008) and allows high productivity in implementing analytical methods, visualisation capabilities are somewhat limited and generating adequate visualisations can be a challenge. However, since data import and export is straightforward, other software packages can be used in conjunction with R to achieve high-quality visual products. Besides the diagnostic toolbox, the prototype for developing and analysing generalisation methods for web and mobile mapping, implements a range of existing generalisation methods from the literature, as well as extending them and creating new ones, and is fully based on Java (see Appendix A).

A more in-depth comparative analysis of the point generalisation algorithm developed in this thesis (see chapter 6 and 7) is presented in chapter 8. Furthermore, the assessment of generalisation algorithms will provide insight into limitations of those algorithms, which can be used for better algorithm parameterisations and will generate ideas for alternative approaches, with the ultimate aim of feeding the results into an automated workflow for point generalisation in web and mobile mapping (see Chapter 9).

Chapter 6.

Object-directed algorithms using hierarchical data-structures

This chapter presents solutions for object-directed generalisation (see Section 3.3.1) of point-data for real-time generalisation. A detailed review of object-directed real-time generalisation algorithms was given in Chapter 2.

Summing up, existing real-time methods either rely on pre-computation and storage in hierarchical data structures, or on generalisation algorithms that are sufficiently efficient that they can achieve real-time performance. While the first approach, as a consequence of pre-computation, lacks flexibility, the latter commonly sacrifices cartographic quality to reduce computational complexity.

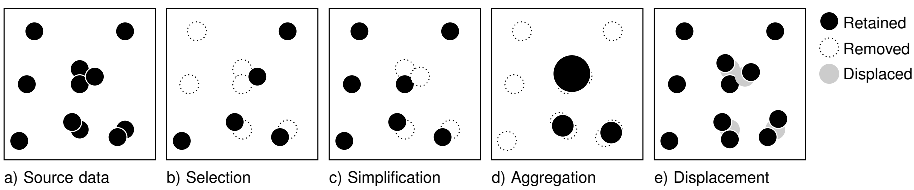


Figure 6.1.: Overview of point generalisation operators.

The presented algorithms in this chapter rely on a point region quadtree as an auxiliary data structure. The application of the quadtree data structures allows the bridging of the dichotomy between approaches based on pre-computation and fast but cartographically suboptimal generalisation algorithms. That is, between flexibility and performance.

Using this index and search structure allows achieving real-time performance for large point sets. The quadtree also supports the computation of various measures such as local point density, and allows the implementation of different generalisation operators (see Section 2.3), including selection, simplification, aggregation, and point feature displacement. Several algorithms are available for each of the generalisation operators summarized in (Table 6.1). Using these different generalisation operators informed by measures computed from the quadtree, mobile mapping applications for point data can achieve flexibility and adaptation. The proposed generalisation algorithms and the creation of the quadtree data structure show real-time performance on a prototype map client as will be

Table 6.1.: Generalisation operators based on the quadtree

Selection	Based solely on feature attributes, applying various selection functions, such as rank, frequency or feature category distribution.
Simplification	Returns one point feature per quadnode, governed by geometric criteria such as centrality, or weighted centrality.
Aggregation	Reduces the number of points by grouping together semantically similar or spatially close points, replacing the original points by a new place holder feature, such as midpoint, clustering, or co-location occurrence.
Displacement	Locally reconfigures point symbols to resolve spatial conflicts by moving points apart from each other. Uses the quadtree for neighbour search.

shown in Chapter 8. The data is not assumed to be known *a priori*. It is important to note that the algorithms are generic and not restricted to be performed solely on a map client.

6.1. Properties of a quadtree

Search operations in large databases quickly become time consuming; therefore they are efficiently supported with index structures, typically in the form of tree structures such as the B-tree (Comer, 1979). Various index structures also exist to efficiently process spatial queries. Known spatial index structures are: clipping techniques, multi-partition techniques, grid-files, R-trees, R*-trees, Range trees, KD-trees and the quadtree (Samet, 2006).

The quadtree is a well known spatial tree data structure, and a generic term for a family of tessellations of the plane in which every node has four children (Samet, 1989). It is well suited for managing point data (Waugh, 1986) and is widely used for 2-D spatial indexing in GIS or other domains, such as in computer graphics for collision detection (Moore and Wilhelms, 1988). Since our case is point data generalisation the point region (PR) quadtree was selected, as it has several properties that make it useful for real-time generalisation. For reviews of quadtrees and associated algorithms, see Samet (1989) or (Samet, 1990).

Properties of quadtrees that are useful for real-time generalisation are listed in Table 6.2. The spatial index speeds up spatial queries and searches. The spatial coverage of quadtree tiles provides information on existence or absence of geographic features in a specified region, and thus enables estimates on feature density/distribution. The topology of quad neighbours provides information about the local neighbourhood structure Samet (1990) to the otherwise unrelated points. And the recursive hierarchical subdivision of map space adapts to point density and thus progressively builds up a spatial hierarchy with implicit scale progression.

6.2. Basic operations on quadtrees

The following section describes a generic algorithm for generalisation, based on the quadtree. It first briefly presents different operations on the quadtree data structure such

Table 6.2.: Properties of quadtrees useful for real-time point generalisation

Property	Use	Generalisation operator
Spatial Index	Speed up spatial search	Selection, aggregation
Coverage	Enables estimates on densities and distribution	Selection, displacement
Topology	Quad neighbourhood	Displacement
Hierarchy	Recursive and progressive subdivision	All operators

as insertion, deletion, get neighbours and querying the quadtree data structure and serves as a base for the different applications which are described later and operations on the quadtree structures for different generalisation operators.

A quadtree can be created by recursively inserting the points in the tree structure. The order of insertion does not define the shape; rather it is highly dependent on the spatial distribution of the inserted point set.

Insertion: The insertion of a point into a point region quadtree happens recursively. A point is inserted inside the leaf node covering the position of that point. If the node is empty the point is stored in that leaf node. If not, the node is further partitioned until there is an empty leaf node where the point can be stored. For two close points many levels of partitioning may be needed. This can be alleviated by allowing more than one object per leaf node, creating a variant of the quadtree also called the bucket quadtree.

Deletion: Deletion of a leaf in a PR quadtree is done recursively, by removing all the children of the leaf's parent node, if all together hold less than one point. This is done, until they together hold more than one point.

Query Range: To find all the points within a specific range in a quadtree, a check is performed on whether the boundary of the quadnode intersects with the query range. In case of intersection a further check is performed, if the stored points lie within the range. If the intersected quadnode has children, the range check is performed recursively if the children contain points that lie within the query range.

Query Range Recursion-restricted: If results of a generalisation operator are stored in the nodes of the quadtree the query range may be restricted to a specified depth, as the level of a tree can be linked to the generalisation level.

Neighbourhood: Finding neighbour(s) in a quadtree corresponds to finding adjacent leaf nodes for each side and corner of a quadnode. Finding neighbours depends on the formulated query and can be restricted to finding neighbours in a specific direction, finding neighbours within a predefined range (distance), or furthermore finding empty neighbours

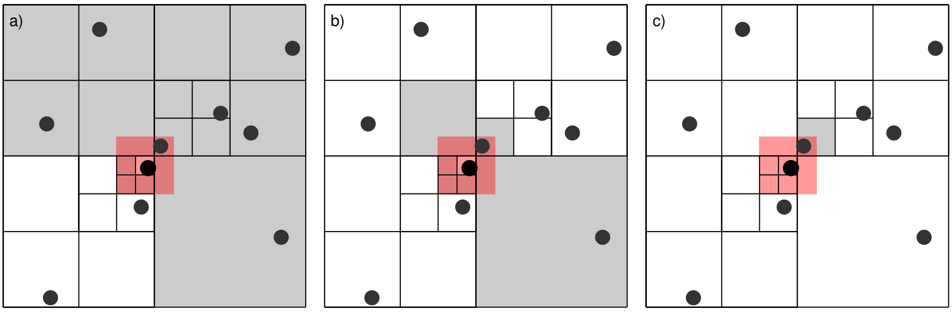


Figure 6.2.: Finding non empty neighbours within a quanode within the same depth of the quadnode (red square). a) Six adjacent parent quadnodes to be searched b) restricted to immediate neighbours c) restricted to non empty quadnodes within the search range

as possible candidates for displacement operations for generalisation. A more in-depth description of neighbour finding methods in quadtrees may be found in Samet (1981).

6.3. Quadtree-based algorithms for point data generalisation

The basic idea of the quadtree-based generalisation approach is to apply generalisation operations to quadtree nodes according to the target *level of detail* (LOD), which is mapped to the level of a quadtree. The basic operations on quadtrees include: *insert*, *delete*, *get neighbours*, and *query* the quadtree (Samet, 1989, 1990). They form the foundation of the generalisation algorithms described below, illustrated in Figure 6.3. The shape of the tree depends on, and therefore reflects, the spatial distribution of the inserted point set and is independent of the insertion order. The quadtree is built in real-time, after the point set is loaded either from a local data repository or via a spatial query to a server (e.g. select restaurants within the greater Zurich area).

The generic flow of our quadtree-based generalisation approach consists of three steps:

1. Creation: In a first step, the quadtree is created in real-time by inserting the loaded point data and their (optional) attributes into the data structure, covering the extent of the query window, and projecting the data from lat/lon coordinates to the coordinate system of the tiling service. Attributes such as feature category, rank (e.g. relevance ranking generated by an external application), or other measures are assigned to the point data, if required by the desired generalisation algorithm. In our implementation, attributes are stored externally to the quadtree and accessed via the point objects' unique identifiers.

2. Generalisation algorithm: In a second step, a generalisation algorithm returns, for each quadnode, the resulting generalized points at the target LOD. The target LOD (and thus target map scale) is mapped to the tree depth. It translates to the width of the quadnode side, measured in screen (pixel) coordinates (Figure 6.4d) and denotes the smallest required distance to resolve spatial conflicts, given cartographic constraints (e.g.

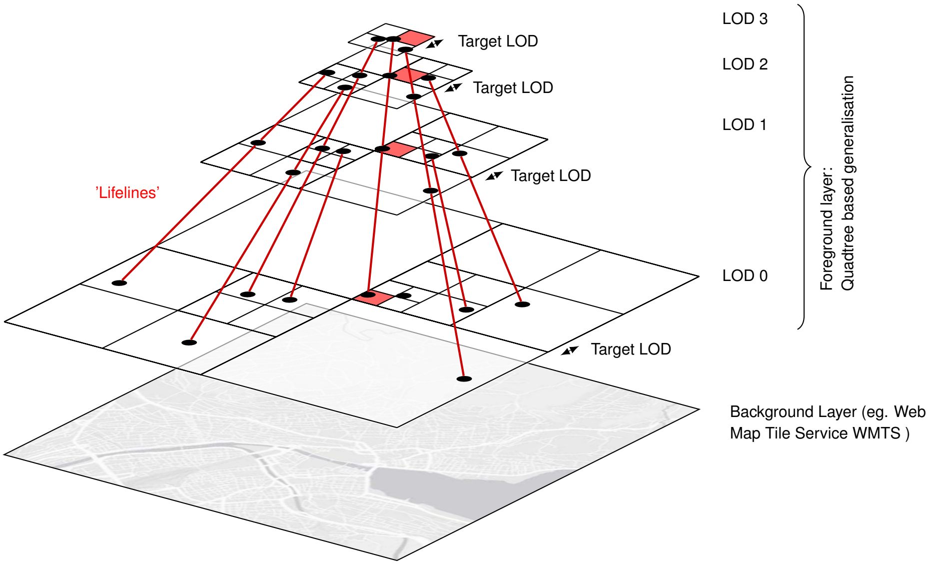


Figure 6.3.: Overview of the quadtree-based generalisation model

point symbol size). Equation 6.1 denotes how the quatree zoom level l_{rel} and the symbol size s relates to the map zoom level z. In other words, the target LOD is defined such that the symbol width is smaller than the width of the quadnode at the target LOD. For each visited quadnode at the desired depth either a generalisation algorithm is applied in real-time to all its child nodes, or the generalized points are retrieved from previous, pre-computed runs (see Section 6.5). In order to fully meet cartographic constraints, point symbols overlapping the border of quadnodes may be displaced if a check of quadnode neighbours reveals that there is an overlap with another point. Details of generalisation algorithms will be shown in the next section.

$$l_{rel} = -\log_2\left(\frac{s}{pw_{bb}}\right)$$
$$l_{zoom} = z - l_{rel} \tag{6.1}$$

l_{rel}	Level relative to quadtree of depth $= 0$
l_{zoom}	Map zoom level with quadtree of depth $= 0$
z	Map zoom level
s	Symbol size
p	Symbol-quadnode ratio
w_{bb}	Screen width of the quadtree bounding box at z

3. Display and caching: In the third step the results from Step 2 are either displayed (display-and-forget), or optionally stored in the nodes for fast retrieval in subsequent iterative generalisation (see Section 6.5).

6.4. Quadtree-based generalisation algorithms

This section presents different quadtree-based generalisation algorithms for the generalisation operators of Figure 6.1 and Table 6.3. The proposed algorithms consist of those that, for a given quadnode, derive generalisation results based solely on the quadnode and its subtree, and those that consider also neighbouring nodes. The second group is computationally more expensive but leads to cartographically superior results. First, however, we briefly review the role of geometrical measures in support of quadtree-based generalisation algorithms.

Measures Geometrical measures help to inform the operation of generalisation algorithms, and parameterise their outcome (see Table 6.3). Local measures are derived from each quadnode and its neighbour nodes, such as the number of points stored in the subtree, maximum depth, size, and balancing of the subtree. Global measures are based on the complete dataset and include global statistics, such as the average number of elements per node and LOD. Such measures can be used to control the generalisation and portrayal process. For instance, the number of points stored in the subtree of the current quadnode at the target LOD can be directly used to adapt the point symbol size in aggregation algorithms and yield graduated symbols (as shown in Figure 6.15). Measures can also be used to set thresholds for the generalisation algorithm. For instance, when pruning elements from the tree preference can be given to quadnodes that represent a large subtree (and thus a large number of points) at target LOD. Thus, the underlying spatial pattern can be retained. The use of measures as parameters in the proposed algorithms thus helps to maintain the balance between local and global maxima of the overall spatial distribution of the data.

6.4.1. Selection

Chooses a subset of points from the original set of points, based solely on attributes, such as a relevance value per point object.

Value-based selection Returns the most significant element per quadnode (Figure 6.4b). 'Most significant' denotes a function of a numeric attribute, such as maximum value of an attribute of points occurring in a subtree. For POI data, such as restaurants (cf. Figure 6.12a), highest relevance may be used (obtained from a relevance ranking service). For point collections (i.e. point data sets that encompass large collections of counts or observation data), such as animal observation data (cf. Figure 6.12b), the most or least frequently occurring observation may be used, or a weight to maintain the overall distribution of counts to avoid over or under-representation of certain feature categories (e.g. ibex in Figure 6.12b). If the numeric attribute of the point is normalized by the depth or the total number of points of the quadnode's subtree, the algorithm will generate a generalisation result that approximates the local point density.

Table 6.3.: Global and local measures applicable to quadtree-based generalisation

Global quadtree measures	
Number of points	Total number of points stored in the quadtree
Depth	Maximum number of subdivisions
Storage capacity	Maximum number of points that can be stored at maximum depth
Average	Average number of points per quadnode at specified depth
Average depth	Average depth of all leaf nodes at specified depth
Occupancy	Number of quadnodes storing points in relation to the storage capacity of the quadtree
Descriptive Statistics	Descriptive statistics of stored point values; minimum, maximum, average, standard deviation, median, mean etc.
Distribution	Distribution of the stored point values (e.g. normal distribution)
Local quadnode-based measures	
Number of points	Number of points stored in the quadnode and its child nodes
Depth	Depth of lowest child node
Has children	Quadnode with child nodes
Occupied / empty	Quadnode (not) storing points
Storage capacity	Maximum number of points that can be stored at a predefined depth
Representation	Number of points compared to average number of points at a specific depth (over/under-representation)
Occupancy	Number of points in relation to total capacity
Density	Number of points per quadnode area
Midpoint	Geometric midpoint of quadnode
Most central point	Most central point of the points stored in the quadnode
Descriptive Statistics	Descriptive statistics and distribution of stored point values; minimum, maximum, average, standard deviation, median, mean etc. (in the case of quadnodes with larger subtrees)
Local statistics	Statistical measures that apply to local neighbourhood (e.g. Morans I).
Cluster node	Quadnode storing more than average number of points or depth higher than average depth with at least one neighbouring cluster node
Cluster reachable	Distance between generalised point and cluster neighbour node is smaller than cluster neighbour node side
Neighbour depth difference	Depth differences between quadnode and neighbouring quadnodes
Neighbour measures	Average number of points stored in quadnode neighbours, number of occupied, empty neighbours and comparison with descriptive statistics of neighbouring quadnodes

6.4.2. Simplification

Like selection, simplification chooses a subset of the original points. However, simplification is governed by geometric properties, as done in line simplification (see Chapter 3 or McMaster and Shea 1992).

Centrality-based simplification Retains the most central point per quadnode, defined by the point lying closest to the mean centre of the points in a quadnode (black dots in Figure 6.4c). Thus, solely requires the positions of the point features.

Weighted centrality-based simplification Either applies the weighted mean by using a numeric attribute to derive the most central point per quadnode, or the central point is obtained by also considering the points of the neighbouring quadnodes, potentially leading to a more balanced generalisation result. Both variants are shown schematically in Figure 6.4d.

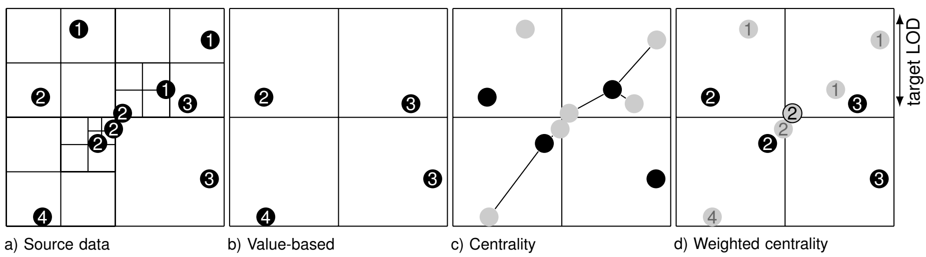


Figure 6.4.: Point reduction algorithms, with results for level 1: a) quadtree source data; b) value-based selection, retaining points with highest attribute value per subtree; c) centrality-based simplification, retaining points closest to the mean centre of all points contained in a quadnode. d) two variants for weighted centrality simplification: central point obtained by weighted mean of attribute values within quadnode only (black dots), vs. central points obtained considering also the neighbouring quadnodes (dots with black outline).

6.4.3. Aggregation

Reduces the number of points by grouping together semantically similar or spatially close points, replacing the original points by a new placeholder feature (i.e. the point positions change).

Quadnode centre aggregation Aggregates points of a quadnode to the position of the corresponding tile centre. This is the most basic of the proposed algorithms.

Midpoint aggregation Aggregates to the midpoint (i.e. mean centre) of the points of a quadnode's subtree (Figure 6.5b). It uses solely the positions of the points. As in weighted centrality simplification, neighbouring quadnodes may be taken into consideration, *e.g.* by weighting the mean centre by the number of points stored in the neighbouring quadnodes.

Cluster-based aggregation Returns a placeholder (e.g. the modal center) for highly clustered and densely populated quadnodes with a large number of children, based on the positions and attributes of the points (Figure 6.5c). Assigns the points to the same cluster if the neighbouring childnodes (four respectively eight connected neighbours) are not empty or yield a minimum, user-defined, number of points. Interesting extensions of this approach are to extend cluster-based aggregation with adapted versions of clustering algorithms, such as STING (Wang et al., 1997) or DBSCAN (Ester et al., 1996).

Co-location filtering Generalizes quadnodes based on a co-location rule, such as the co-occurrence of features in a quadnode belonging to the same class, or capturing logical relationships between different point categories (Figure 6.5d). Co-location of one or more point features in space seems to influence the user's decision in location-based services (Reichenbacher and De Sabbata, 2011). An example for co-location is finding co-located parking lots and restaurants in a city. For each quadnode's subtree the co-location algorithm checks if it contains restaurants and parking lots, and if so, how often, in order to scale the resulting point symbols proportionally to the number of co-occurrences of restaurants and parking spaces within that node resulting in one aggregated symbol (Figure 6.5d) or two separate symbols (Figure 6.5b) per generalized quadnode. To avoid edge effects from not considering co-located elements across boundaries of two neighbouring quadnodes, the algorithm can be extended to check for co-location also in neighbouring nodes.

Note that while co-location filtering conceptually might be thought to be a selection operation, it is counted among the aggregation algorithms, since it aggregates multiple co-occurrences within a quadnode and its subtree.

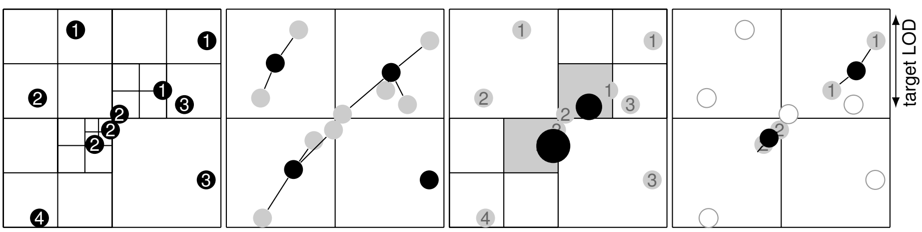


Figure 6.5.: Aggregation algorithms: a) quadtree source data, b) midpoint aggregation; c) cluster-based aggregation; d) co-location filtering for co-occurring same valued points.

6.4.4. Displacement

Locally reconfigures point symbols to resolve spatial conflicts by moving points represented. It is thus usually applied for the 'finishing touches', as the last in the chain of generalisation operators. Its application is limited to the resolution of spatial conflicts in a narrow band of scales, typically from one LOD to the subsequent one. Our quadtree-based displacement algorithm acts – as the previously presented quadtree algorithms – on the target LOD. The algorithm reallocates those points of a quadnode not satisfying cartographic proximity constraints (framed red in Figure 6.6) to neighbouring quadnodes.

Points can be displaced to their cardinal neighbours depending on whether the king's or rook's move is applied (Figure 6.6a, b). Therefore for each point stored inside the four childnodes there are two (rook's case) or three (king's case) favourable displacement options. The actual direction in which a point is displaced is preferably directly opposed to the direction of the mean centre formed by all the quadnodes points (Figure 6.6e). The preferred range of directions in which a point can be displaced is limited by the bisectors of its immediate neighbouring points to the mean centre, and by the holding capacity of the neighbour nodes.

A point can only be displaced to the neighbouring quadnode if that node shares the same depth and is empty, or has a lower depth (grey shaded quadnodes in Figure 6.6d). This is controlled by the quadnodes' holding capacity: it indicates how many points the neighbouring quadnode can possibly contain, without further subdivision (Figure 6.6d). For each point the holding capacity for each possible displacement direction is summed up and the points are sorted in ascending order (Figure 6.6c). The point closest to the midpoint of the points is set to the end of the list. The childnode with the least possibilities for displacement is displaced first, reducing that neighbour's holding capacity by one. If there is no other further possibility of displacement the point is kept. If the neighbouring quadnode is of a depth less than the current quadnode and already holds one point, this node will be further subdivided to insert the moved point (Point C in Figure 6.6f). The subdivision, however, will not exceed the target LOD. If after subdivision both points happen to be inside the same childnode, the moved point is discarded. The goal is, at the limit of display resolution (i.e. at the target LOD), to have one or zero points per node.

If no more points can be moved the algorithm stops and keeps the most central point (Point A in Figure 6.6f) or the one with the highest importance value, and removes the childnodes and remaining points that could not be displaced.

Extension The above algorithm can be extended by allowing point displacement to be propagated to neighbours of degree 2. If a point cannot be displaced to its immediate (degree 1) neighbours using the above algorithm, the algorithm would try to displace an immediate neighbour in the preferred direction of displacement first, to make room for displacement. For performance reasons, it is advisable to limit the search to a predefined, small degree of neighbourhood.

6.5. Caching

The above algorithms are designed for real-time performance and no caching (i.e. temporary storage) is needed (display-and-forget mode of operation). However, the quadtree

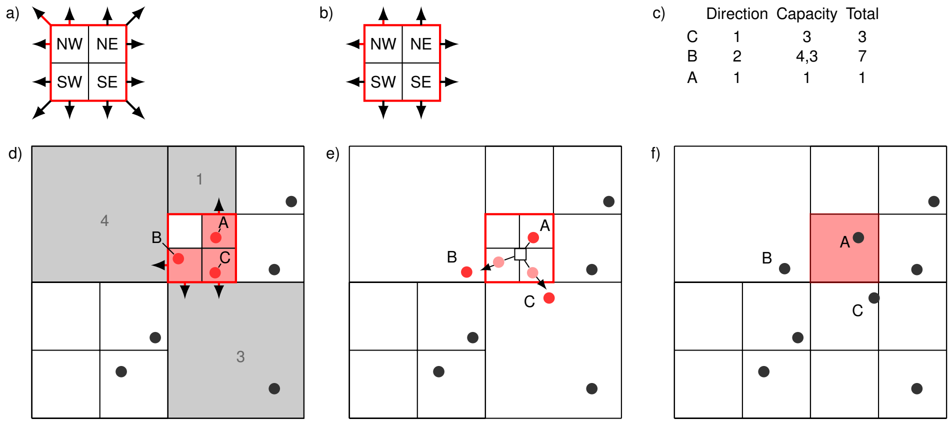


Figure 6.6.: Steps applied by the displacement algorithm for the red quadnode: a, b) movement options per child quadnode (rook's move vs. king's move). c) sorted table of summed holding capacity per point. d) Points A, B, C stored in the childnodes are candidates for displacement. Gray shaded areas denote candidate target quadnodes for displacement, numerals denote their holding capacity. Red arrows indicate the possible movement direction for the points contained in childnodes. e) Optimal displacement directions for B and C. A is kept, as it is the most central point. f) B and C are displaced and the childnodes removed.

also lends itself nicely as a caching structure for generalisation results (rather than serving merely as spatial index). Caching of the generalisation results directly in the nodes of the quadtree facilitates faster interactive zooming, as long as the generalisation algorithm and/or the search criteria do not change. If the application requires frequent change of generalisation algorithms, the result may be stored using identifiers for the respective generalisation result that are stored in the quadnodes.

Figure 6.7 schematically illustrates the use of the quadtree as a caching structure. In display-and-forget mode (i.e. the normal case), the generalisation algorithm returns the results for each visited node and its subtree, but does not store it persistently (Figure 6.7a shows an example for LOD 1). However, in the same pass, the results of the generalisation algorithm can also be stored in each quadnode visited. If results have been cached and if, due to zooming, the same LOD is requested a second time, the portrayal process can simply retrieve the cached result from each quadnode at the requested LOD, optimizing the portrayal process in terms of speed. Figures 6.7b and 6.7c show two different ways of exploiting the caching tree for retrieval. Figure 6.7b depicts the case when zooming out: for an LOD k that is visited in the ascending order, only the previous LOD $k + 1$ needs to be retrieved to derive the full content of the LOD k. Figure 6.7c depicts the usage case typical of generating approximate generalisation results. The search range per quadnode is restricted to a user-defined depth $k + i$, which depends on the generalisation algorithm applied. For instance, for a displacement algorithm it is sufficient to only consider a few

LODs (e.g. LOD $k + 1$), as displacement is an operation that acts only locally.

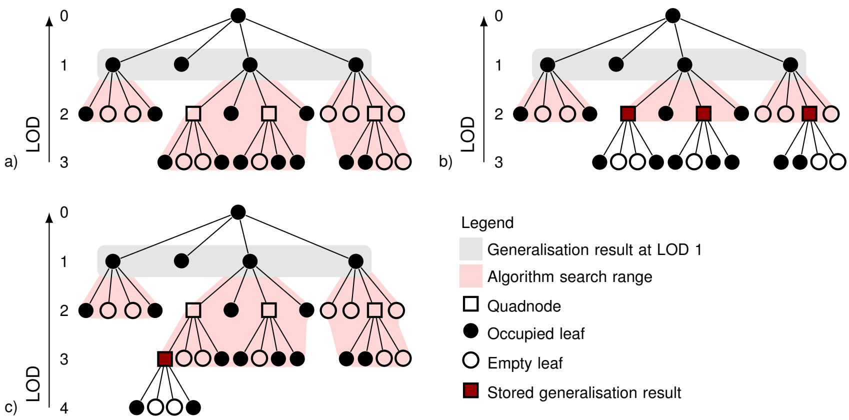


Figure 6.7.: Retrieval from cache for generalisation at LOD 1: a) for each quadnode its complete subtree is considered (equivalent to display-and-forget mode, i.e. no caching); b) for each quadnode at LOD k $(= 1)$ its leaves and the generalisation results (black squares with red fill) of the LOD $k + 1$ $(= 2)$ are considered; c) depth restricted retrieval with LOD $k + i$; all leaves up to $k + i$ $(= 3)$ and the generalisation results of $k + i$ $(= 3)$ are considered.

6.6. Foreground-foreground constraints

Conflict constraints: The described quadtree-based generalisation operators do not *a priori* resolve all the overlaps between the generalised points, that is, between foreground and foreground objects. Overlap to a certain extent may still occur along the border of the quadnodes. By adding a check for overlaps with generalised points these remaining overlaps are fully removed. The removal of the remaining overlaps comes with a small performance cost due to the lookup of the neighbour point conflicts. By directly storing a reference list of the neighbouring nodes inside each quadnode of the quadtree, the querying of the neighbouring quadnodes is avoided.

The two main conflict resolution strategies are move and avoid; move is the conflict resolution strategy for content-conservative generalisation operators, while avoid is applied with space-conservative operators.

There are several ways in which conflict constraints can be addressed. Border conflict constraint resolution options are *none*, *move inside*, *avoid*, *move*, and *fill*. The different resolution types are shown in Figure 6.8. *None* is the case if no conflict resolution is requested. *Move inside* moves all points inside each node, regardless of whether there is a conflict constraint violated or not. This resolves the conflict constraints but also introduces a certain regularity which is visible depending on the underlying spatial point pattern. *Avoid* retains one of the overlapping points, depending on the applied selection criteria. *Move* displaces conflicting points to a non-conflicting position. *Fill*, as a fourth

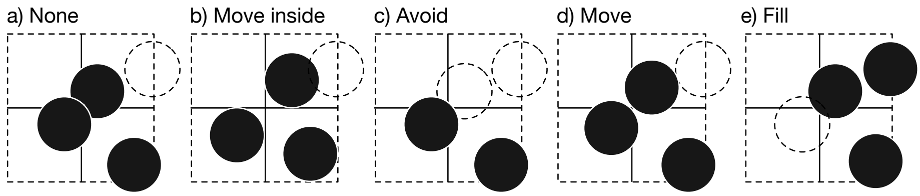


Figure 6.8.: Border conflict resolution options (conflict constraints applied to the four or eight cardinal directions)

option, is a special conflict constraint resolver. As the name says, it fills up the node with points stored inside a quadnode, such that there are no conflicts. It only allows to show a point at a specific zoom level if the point is not in conflict with already selected points, instead of first selecting the generalised point per quadnode and then checking for conflicts as done in *avoid* and *move*. With this conflict resolution method a quadnode may contain more than one representative at a specific zoom level.

The application of conflict constraints also addresses seemingly 'fake clusters'. These are formed by two close points across quadnode borders with no child nodes, visually forming the impression of a cluster. Conflict constraints and a depth check removes 'fake clusters'.

Grouping constraints: Grouping constraints allow the emphasis of clusters on generalised maps. Due to the generalisation process more points are generalised in dense areas and hardly any points need generalisation in areas of low point density. In areas of high point density, due to conflict resolution, these areas appear less dense on a generalised map. In contrast, areas with a low density show in relation to areas of high density a seemingly higher density. Hence, if the generalised map should emphasize the coverage or the local and global clusters, grouping constraints can be applied (see Figure 6.9). Grouping constraints are well suited for coarse map zoom levels.

The *depth grouping constraint* d is based on the maximum depth reached by the leaf nodes of a quadnode. This constraint only allows points to be kept if a quadnode that contains points has a maximum depth that is at least the depth required by the current map zoom level, added to the constraint value d.

The *point-based grouping constraint* p sets the requirement such that the points stored in a quadnode are only retained for generalisation if the number of points is at least the number of points defined in the grouping constraint p.

6.7. Foreground-background constraints

While all solutions to resolve foreground-foreground conflicts have been implemented, this section merely proposes conceptual solutions to deal with conflicts between background and foreground objects. The use of a quadtree as a spatial index infers relationships among otherwise unrelated points. That is, it allows the formation of a topological

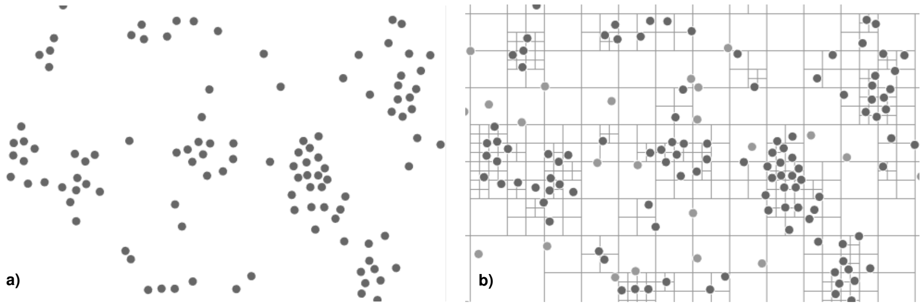


Figure 6.9.: a) Result of value-based selection with applied grouping constraints (depth grouping constraint); b) Debug-view - dark points denote retained points, points in light gray have been removed by the grouping constraint settings.

structure and a coverage, which associates a *quadnode region* to the points. The region a quadnode covers (and its close neighbourhood) permits the formulation of background-foreground constraints, defined in Section 3.1.1. Foreground-background constraints mainly play a role if the generalisation operator displaces points from their original position, such as in the case of typification, aggregation and displacement operators.

Figure 6.10 shows how foreground-background constraints are addressed by using the the quadnode as a *clip region* to infer foreground-background constraints based on the quadtree foreground layer and background *constraint layer*. The usage of background constraints strongly depends on the background data model and current scale; that is, the map scale of the data should be sufficiently close to the target scale. Secondly, depending on which data model (raster or vector) is used in the constraint layer, the methods for considering background-foreground constraints differ.

The resolution of foreground-background constraints can be described as follows: define and select the constraint layer, apply quadnode region clipping, prepare the clip region to form a constraint region, and lastly resolve background constraint violations (see workflow in Figure 6.10):

1. Constraint layer types: Depending on the data model used in the constraint layer, two main constraint layer types exist, *vector-based* or *raster-based* constraint layers. Typically the background layer forming the visual backdrop for web and mobile mapping is a raster-based solution, such as a web map tile service (WMTS) provided by Google maps, Bing maps and the like. To account for background constraints, the constraint layer is not necessarily the one used in the final map as the background layer.

Depending on the map objects described in the constraint layer, such as lakes, roads and classified pixels of satellite images, various background constraints can be formulated ranging from rather generic – if the semantic of the background objects are unknown (unclassified pixels or simple lines) – to rather specific, with clear semantics attributed to the map objects. For instance, a layer with all lakes and rivers may be used as a mask to avoid moving map objects into lakes, or a train network may be used in order to avoid

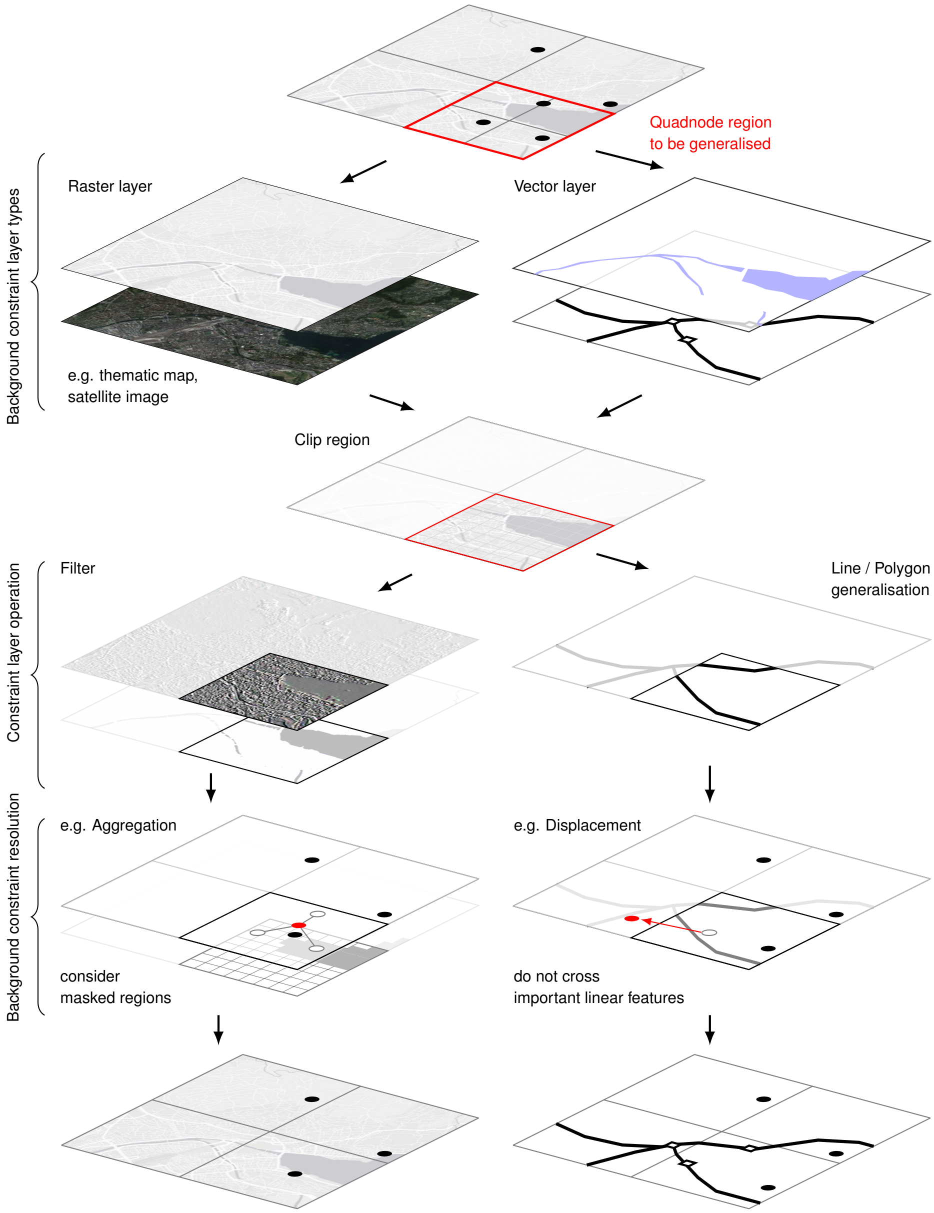


Figure 6.10.: Quadtree-based generalisation with background constraints. The sketched workflow illustrates based on two examples – aggregation (left side) and displacement (right side) – how different constraint layer types (raster and vector layer) can be applied to implement foreground-background constraints.

moving map objects across major geographic features.

2. Quadnode region clipping: Defines the clip region within the constraint layer; typically the region that comprises the quadnode to be generalised, or, depending on the generalisation algorithm, also its close neighbourhood. The next steps then consider solely the data inside the clip region.

3. Constraint layer operations: Depending on the constraint layer type, raster or vector, the data of the constraint layer is filtered and generalised with basic algorithms to prepare the data for resolving the generalisation constraints. In the case of a *raster-based constraint layer*, filtering and pixel classification operations are applied to create lookup maps and masks, and derive patterns such as linear patterns through edge-detection (e.g. Sobel filter) or pattern-detection by quadtree image decomposition (Samet, 1984) (see also McMaster and Shea 1992 and Schylberg 1993).

Vector-based constraint layers ideally need no further generalisation to form the constraint layer, if simple reduction or simplification algorithms are used. It should be noted, however, that due to geometric changes the topology between foreground and background elements eventually changes.

4. Background constraint resolution: If the quadtree-generalisation method assigns a new position to a point feature, a check against background constraint violation is executed, as shown in Figure 6.11. If background constraints are violated new non-violating positions are sought or a point reduction algorithm is applied. Non-violating positions are ranked by their closeness to the point feature's original position. Another approach to consider is the concept of *least disturbing place* and the algorithm described by Harrie et al. (2004).

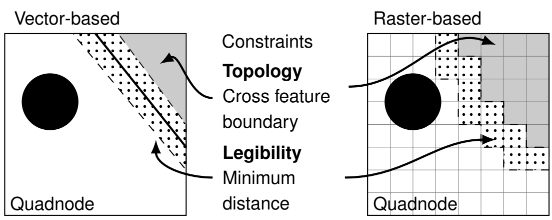


Figure 6.11.: Background constraints for vector and raster-based background constraint layers

Figure 6.10 exemplifies this step, based on an example involving an aggregation and a displacement algorithm. Constraint rules are predefined. The two examples with background constraint violation illustrate both cases. In the aggregation example (left side), a new position outside the lake mask – formed by a raster-based constraint layer – is selected. In the displacement case, the point is removed and not moved to the neighbouring quadnode, as the point eventually crossed an important linear feature.

The formulation of background constraint rules depends on the availability and quality of the data. Useful constraint rules for web and mobile mapping are: *proximity*, to maintain minimal distance to background features and maintain topology, or *object relations* – that is, to maintain containment and alignment of foreground against background map objects (e.g. do not cross important linear features) see also (AGENT, 1998; Beard, 1991; Steiniger and Weibel, 2007; Burghardt et al., 2007).

6.8. Prototype and data

The proposed quadtree-based generalisation algorithms have been implemented in a prototype generalisation platform using Java and Processing (www.processing.org). Appendix A provides an in-depth description of the prototype. Two datasets were used for the foreground data: a POI dataset of eating places in the Canton of Zurich, Switzerland, originating from OpenStreetMap (Figure 6.12a); and a point collection of cumulative animal observation counts provided by the Swiss National Park (Figure 6.12b). Further details on the two datasets are provided in Appendix B.

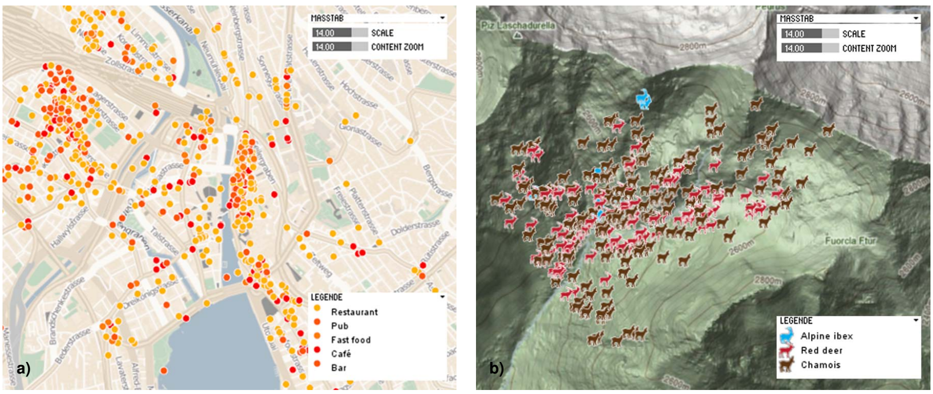


Figure 6.12.: a) OSM dataset POI in Zurich, Switzerland depicting eating places by type. b) Point collection of animal observation data depicting animal types, courtesy Swiss National Park. Base map: a) ©2012 CloudMade – map data CC BY SA 2012 OpenStreetMap.org, b) ©2012 Google

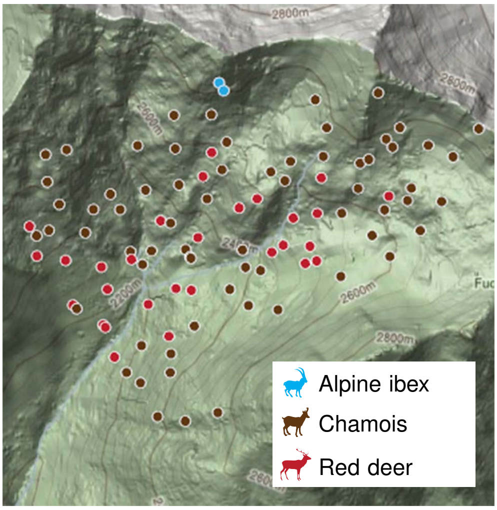
The POI dataset features several categories, from which a subset (restaurants, bars, cafés etc.) was extracted as a thematic layer for this paper. The animal data are cumulative and represent point collection; that is, the points represent observation counts at point locations, ranging from individual animals to entire flocks.

6.9. Experimental results and discussion

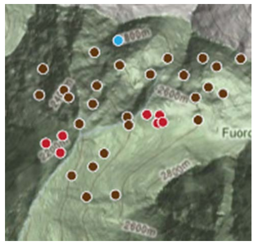
This section presents results generated by some of the quadtree-based generalisation algorithms described above.

Selection based on ranking, or a selection function based on attributes only retains one representing element per quadnode. In Figure 6.13 only the point representing the highest animal observation count per quadnode is kept for a given LOD. Due to the strong selection confined to quadtree tiles, most of the spatial conflicts are resolved, though no displacement to resolve remaining overlaps was applied. The progression through LODs 14 to 12 shows how the generalisation retains the spatial point configuration relatively well. Concerning the number of points retained, our algorithm differs quite strongly from Radical Law (Töpfer and Pillewizer, 1966). An in-depth description of the data reduction characteristics of the quadtree-based generalisation algorithms is presented in Section 8.5.

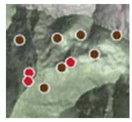
LOD 14



LOD 13



LOD 12



LOD	Scale	Points	Radical Law
14	1:36000	98	98
13	1:72000	34	69
12	1:144000	10	49

Figure 6.13.: Value-based selection for LOD 14, 13, 12. For each quadnode, the algorithm retains the point with the highest observation count. Base map: ©2012 Google

While value-based selection retains the values a user is most interested in, simplification by centrality keeps the most central point per quadnode (Figure 6.14), that is, the point closest to the mean centre of all points falling within the generalized quadnode. Figure 6.14b illustrates how the most central point is selected. As becomes noticeable, the selection of the most central point becomes more stable when more points are contained in a quadnode, whereas it often generates questionable results for quadnodes containing only two points. Furthermore, clusters that consist of many points but happen to fall into the same quadnode are reduced to a single point, while single points that happen to fall into neighbouring quads will persist as a group of four. This could be alleviated by using graduated symbols or by an extension of a border conflict constraint to add an extended distance threshold for this special configuration. However, it requires the definition of a visual separation distance, based on the quadnode width and a selection rule.

Aggregation groups two or more points and replaces them with a new place holder point feature, with the point symbols optionally scaled by the number of points aggregated to generate a graduated symbol map. Note that proportional symbol size can be used as an

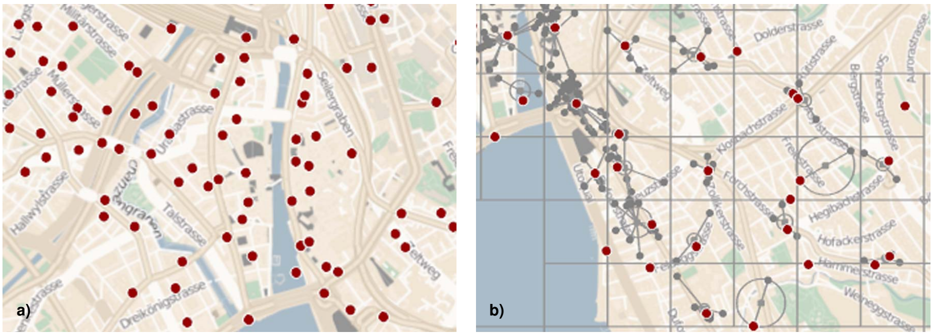


Figure 6.14.: a) Centrality-based simplification for LOD 14; b) enlarged debugging view: rectangles depict quadnodes, gray dots removed points, the gray square symbols denote the mean centre of the points inside each quadnode, and the circle shows the distance to the closest point from the mean centre. Base map: ©2012 CloudMade – map data CC BY SA 2012 OpenStreetMap.org

additional cartographic option with all of the algorithms proposed above (not just aggregation), thus adding further information about the density of underlying original points. Figure 6.15a shows the result of midpoint aggregation, as blue, graduated circles proportional in size to the number of aggregated points. For comparison, the points resulting from ranking-based selection for the same LOD are overlaid. As can be seen, aggregation and selection are equivalent in areas of low point density, where quadnodes at the target LOD only contain a single point.

Co-location filtering is another type of aggregation where spatially co-occurring points are retained. Again, the map symbols depicting the resulting points may be weighted by the frequency of co-occurrence. Border effects – that is, close elements within two neighbouring quadnodes – can be accounted for by a check in neighbouring quadnodes of nodes with no occurring co-location. Figure 6.15b shows aggregation by co-location filtering, with symbol size proportional to the number of co-occurrences of restaurants and parking lots per quadnode.

To further resolve spatial conflicts in a map and/or retain more elements than, for instance, in the case of centrality-based simplification, a displacement algorithm can be applied (Figure 6.16). The displacement algorithm tries to accommodate as many points as possible, keeping at most one point per quadnode. Thus, it denotes the highest possible holding capacity for the current LOD given cartographic legibility constraints. On the other hand, it also homogenizes dense clusters and thus affects the overall distribution pattern (compare Figure 6.15a and 6.15b). The proposed displacement algorithm only displaces a point to the neighbour of a quadnode if the neighbouring quadnode has a holding capacity higher than one for the current LOD. If, for a particular quadnode, no displacement is needed, no check is applied for further potential overlaps with points contained in quadnodes adjacent to the node's boundary. Thus, some overlaps are still visible in Figure 6.16b, which could be resolved by an additional check while applying the displacement

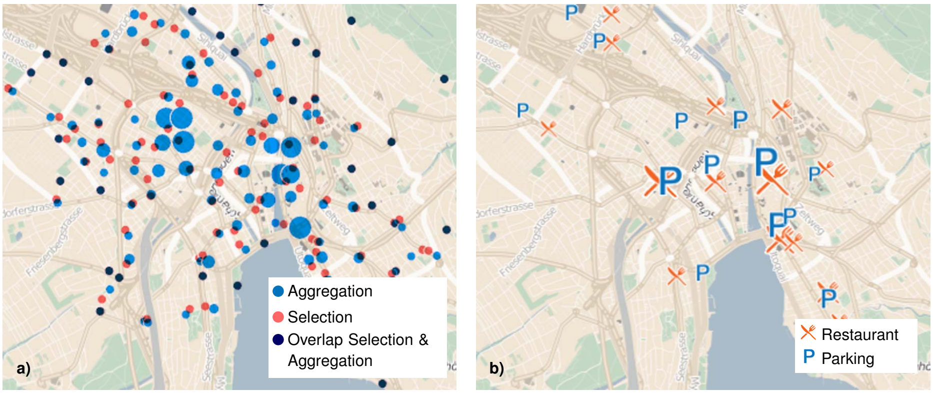


Figure 6.15.: a) overlay of midpoint aggregation, with symbol size weighted by number of aggregated points), and value-based selection for LOD 13, overlaps shown in dark blue; b) aggregation by co-location, with symbol size weighted by the number of co-occurrences for restaurant and parking lot. Base map: ©2012 CloudMade – map data CC BY SA 2012 OpenStreetMap.org

algorithm. However, the displacement result (Figure 6.16b) is capable of displaying more points than the corresponding centrality-based simplification (Figure 6.16a), since it was possible to remove some overlaps and move points to adjacent quadnodes. In general, for all algorithms that do not maintain the original point positions (i.e. in aggregation, displacement operations), points may be moved to positions that violate possible constraints of the background map (e.g. restaurants may fall inside a lake). The maximum displacement distance of an aggregated or displaced point is half the diagonal of a quadnode at the target LOD and therefore linked to the selected symbol size. Thus, the maximum possible displacement can be controlled. Furthermore, including background constraints and adding a mask layer of positions to avoid (e.g. a water body mask for the restaurants) would remove that issue but also make the algorithm less generic.

6.10. Concluding remarks

Real-time generalisation of point data is required to legibly display foreground data in online map services for web and mobile mapping, such as web map mashups and LBS. This chapter showed that the quadtree data structure offers a very useful framework for designing generalisation algorithms for point data that are capable of operating in real-time, including the construction of the PR quadtree. A comprehensive set of quadtree-based algorithms was proposed that implement the major generalisation operators on point data: *selection*, *simplification*, *aggregation*, and *displacement*.

In addition this chapter illustrated how the quadtree data structure can additionally be used as a caching structure. The experiments conducted in this study demonstrate the application of the generalisation algorithms, using two very different datasets of POI data and point collections. The selected set of experiments shows the potential of using the

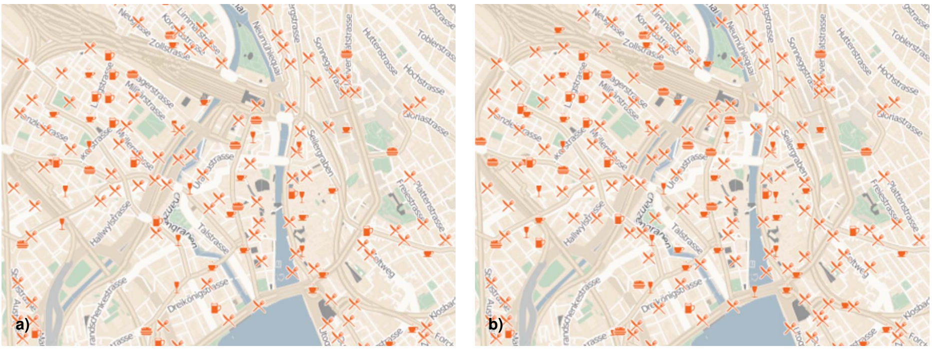


Figure 6.16.: a) Result of centrality-based simplification, displaying 138 elements; b) displacement applied after simplification based on centrality, displaying 184 elements. Base map: ©2012 CloudMade – map data CC BY SA 2012 OpenStreetMap.org

quadtree for real-time point data generalisation. A more systematic set of experiments will be shown in Chapter 8, assessing more thoroughly the strengths and weaknesses of the proposed algorithms, and comparing them to 'classical' generalisation algorithms.

Chapter 7.

Space-directed algorithms using hierarchical data-structures

Chapter 3 on the methodology and Chapter 2 on the state of the art, respectively, present space-directed approaches to point generalisation as a complementing approach to object-directed generalisation. This chapter aims to discuss two selected solutions with a set of experiments, and shows how these fit into the overarching methodology of quadtree-based real-time point generalisation.

In the conceptual classification of point generalisation algorithms (see Figure 3.1 on page 27) the space-directed transformation focus distinguishes two groups, *focal transformations* (e.g. fisheye or polyfocal transformation) and the concept of *malleable space*.

The fundamental difference between the two groups is, what principle triggers the transformation of the map space. The group of focal transformations deforms the underlying map space at one or several predefined regions on the map, based on the metaphor of lenses, whereas the concept of malleable space – a data driven approach – transforms locally, depending on the map objects, the underlying map space. Hence, the two groups are distinguished by whether the transformation of the space is *focus-driven* or *data-driven*.

This chapter concentrates mainly on the concept of malleable space, as it shares the characteristics of being data-driven, with the quadtree-based generalisation solution on object-directed generalisation discussed in Chapter 6. Space-directed approaches are only sparsely discussed in the literature on map generalisation (Edwardes et al., 2005). For a literature review on space-directed algorithms see Sections 2.6, 3.3.2 and 4.1.1.

7.1. Space-directed generalisation of point data

Variable scale maps have been advocated by several authors in the context of mobile mapping. A key advantage is that the topological relations between foreground and background information are maintained after the deformation of the map space.

By deforming the map space, space-directed map generalisation seeks to resolve cartographic conflicts among map features. The two presented methods – the diffusion-based cartogram by Gastner and Newman (2004) and the Laplacian Smoothing approach by Edwardes et al. (2005; 2007) – nicely represent the group of malleable space algorithms. These two approaches, following the concept of malleable space presented in this chapter, transform the map space, that is the map background together with the map foreground.

Similarly to displacement as an object-directed generalisation operator, space-directed generalisation is typically applied within a limited range of scales, optimally at the scale which is best suited to study the selected phenomena. Depending on the data and the map task for small map scales it may not be necessary to use the space-directed generalisation, as the data is too scarce, and no distortion is needed to resolve clutter. For a large map scale, however a satisfactory reduction of clutter often cannot be achieved with space deformation alone. Hence, object-directed algorithms have to be applied to improve readability, or a combination of both object- and space-directed algorithms.

The quadtree can be used as a *'binning' structure* to guide the parameterisation of the space-directed generalisation methods and therefore as a means of controlling and limiting the map deformation. Additionally, quadtree as a *spatial index* provides a tool to combine object- and space-directed generalisation. A combination of the two transformation foci allows, at small map scales, the maintenance of details in dense areas and data reduction in sparse areas.

As will be shown below space-directed transformation foci – *focus-driven* or *data-driven* – combined with quadtree-based object-directed generalisation enable to show more detail in regions of interest or high density, while increasing the degree of generalisation in less interesting regions and regions of low density.

Space-directed generalisation transforms of the background features together with the foreground features. The two methods presented in this chapter are based on a grid structure. The deformation of the grid allows to map the background tiles on the deformed grid structures (same method as used in texturing 3D objects). The new position of the foreground points on the deformed map is calculated based on the known deformation of the background map and projects the foreground features to the correct position.

7.2. Cartograms

7.2.1. Working principle

Cartograms denote a type of map, where the area of geographic regions is drawn in proportion to a variable, for instance the population of a country. The construction of such cartograms is quite challenging and a multitude of algorithms exist, such as the rubber map method by Tobler (1973), Dorlings non-contiguous cartograms (Dorling, 1996) or the diffusion-based method by Gastner and Newman (2004) and others (Dougenik et al., 1985; Tobler, 1986; Gusein-Zade and Tikunov, 1993; Kocmoud and House, 1998b,a; Keim et al., 2004).

The construction of cartograms requires finding a transformation from the untransformed to the transformed map, such that the areas on the map are proportional to a given statistical variable; hence the synonym 'density equalizing maps' for cartograms.

The diffusion-based method for producing density-based maps by Gastner and Newman (2004) avoids the drawbacks of previous methods of constructing cartograms. The diffusion cartogram, as they call it, is based on the observation that in a 'true' cartogram, after the transformation the density of the statistical variable such as the 'population' is uniform. Thus, in regions with high density the variable 'population' has to flow to regions of low density until the distribution of the variable 'population' is equalised. The

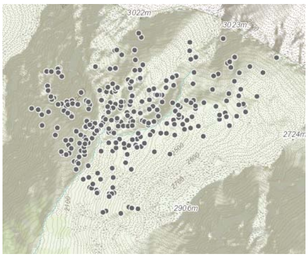


Figure 7.1: Test region for the space-directed point data generalisation with animal observation counts in the Val F'tur in the Swiss National Park (SNP) shown at map zoom level 14. Base map: ©2013 ESRI ArcGIS Services

method by Gastner and Newman adapts the linear diffusion process of particle physics to model the cartogram transformation.

> Part of the art of making a good cartogram lies in shrewd decisions about the definition of the population density. (Gastner and Newman, 2004)

The working principle of the method, however, is only half the story, as the authors state. The method requires the definition of a starting density for the map. That is, the statistical variable 'population' needs to be transformed into a 'continuous function', a density field over the map space represented by a fine grid in their implementation. There are various ways to define the initial density. Depending on the initial density the resulting cartograms may differ quite substantially from each other. Therefore, the initial density surface affects the accurate density equalisation and the readability of the map.

Figure 7.2.: Initial point density grids for the test region of Figure 7.1, using diffusion-based cartogram method by Gastner and Newman (2004) of dimension 64 x 64. a) regular grid with points per cell density b) kernel density estimation with the same grid size using a biweight kernel and bandwidth estimation based on Silverman (1986) c) quadtree derived grid with density estimation for each quadnode

Gastner and Newman (2004) observe that for fine-grained density distributions, cluster centres show considerable distortion. Coarse-grained density distributions cause less distortion, but result in a less accurate density equalisation. Furthermore strongly varying density distributions increase the processing time of the algorithm and special care has to be given to regions with 'zero' density as the method can not handle density values of zero. Gastner and Newman (2004) add a small constant to the density to circumvent this problem.

In applying cartograms as a method for space-directed point generalisation, the data points serve as the basis for the initial density distribution. The transformation of the point data into the starting density in a 'binned' surface for the cartogram can be achieved through different methods (Figure 7.2). One method is by creating a *regular grid* and adding the value of the point to its containing cell, another by creating a *kernel density estimation* (KDE) of the dataset generating a smooth density surface, and a third method uses the *quadtree* and adds the value of the points to its containing quadnode.

The use of a regular grid generates quite substantial distortions, as the values between the cells show strong density variations. The strong variations can be mitigated through the use of a KDE, exploiting its smoothing properties. A similar effect can be achieved by using the quadtree as a binning structure and as a density approximation of the dataset, since the quadtree decomposition adapts to the point density. The three different density grids are shown in Figure 7.2 with subfigure a) depicting the regular grid, subfigure b) illustrating the KDE for the test region are and c) showing the quadtree based grid. The quadtree shown in subfigure c) is drawn up to the smallest and deepest quadnode. The nodes shown here are limited with regard to the current map zoom level (target LOD) and symbol size used in the same way as introduced in Section 6.3 on page 72. That is, the width of the smallest quadnode shown is just equal to, or greater than, the symbol width.

The data for this experiment (for further details see Appendix B), shown untransformed in Figure 7.1, originate from an animal observation dataset from the Val F'tur in the Swiss National Park (SNP). The cartogram implementation used here is a modified version of the implementation ScapeToad by Christian Kaiser[1].

Figure 7.3 shows the resulting cartogram with three different density grids used. Subfigure a) is based on the KDE grid shown in Figure 7.2b), while two different quadtree grids were applied in subfigure b) and c). Subfigure b) shows a quadtree density grid (shown in Figure 7.2b) that records the ratio between the number of elements per quadnode to the quadnode area (i.e. point density), while in subfigure c) the density grid stores, for each quadnode, the number of elements. Note that for all cells in all three density grids a small offset constant is used to avoid that grid cells store density values of zero.

Compared to the two quadtree-based solutions the KDE-based cartogram (Figure 7.3a) results in greater distortion, especially in the dense centre of the valley. The original shape of the point pattern is the least preserved. The distortion by the quadtree-based solutions is less strong. The different weighting schemes in subfigure b and c, respectively, show that even though the same spatial subdivision is used, a slightly different weighting considerably influences the spatial distortion of the cartogram.

That is, the selected target level that affects the size of quadnodes and the values set for the quadnodes in the distortion grid, has to be selected such that it maintains the distortion – and therefore the readability – of the map. Similarly, as discussed in Chapter 4 on content zooming, the user can change the target zoom level for the cartogram grid

[1]ScapeToad, the cartogram implementation by Christian Kaiser is a Java adaptation of Mark Newman's C code. Software: http://scapetoad.choros.ch Source code: https://github.com/christiankaiser/ScapeToad

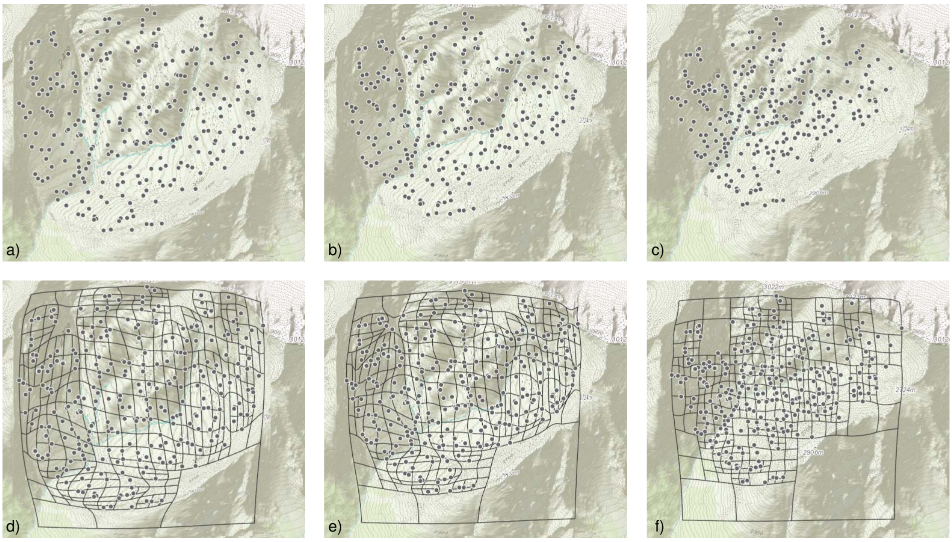


Figure 7.3.: Cartograms generated using different initial density grids: a) KDE grid (see Figure 7.2b) b) Quadtree grid with point density c) Quadtree grid with point count (see Figure 7.2c). d-f) show the same maps as in a-c) but with the quadtree overlaid to better highlight the map distortion. Base map: ©2013 ESRI ArcGIS Services

and change the distortion of the cartogram.

7.2.2. Cartograms – zoom dependent effects of the density grid

Figure 7.4 shows the cartogram with an initial quadtree density grid, generated using the point density method, for zoom levels 14 – 12. The distorted quadtree grid is overlaid on top of the map to highlight the map deformation and the binning structure used to shape the cartogram.

Table 7.1.: Descriptive statistics for cartogram displacement vectors with different quadtree density grids for quadtree zoom levels 14 – 12, and compared to KDE (Figure 7.2b) and a regular density grid (Figure 7.2a)

Map zoom level	14	13	12	KDE	Grid
Min	1.99	2.37	8.91	6.4	3.76
Max	90.77	80.27	80.45	120.4	230.72
Mean	49.18	45.87	46.02	62.34	88.12
Standard deviation	19.05	17.53	16.65	26.96	45.27
Sum	14114.91	13164.99	13209.17	35786.31	50582.28

Numerically the characteristics of spatial deformation can be shown by plotting the cumulative displacement vectors (in number of pixels) per direction, and their descriptive

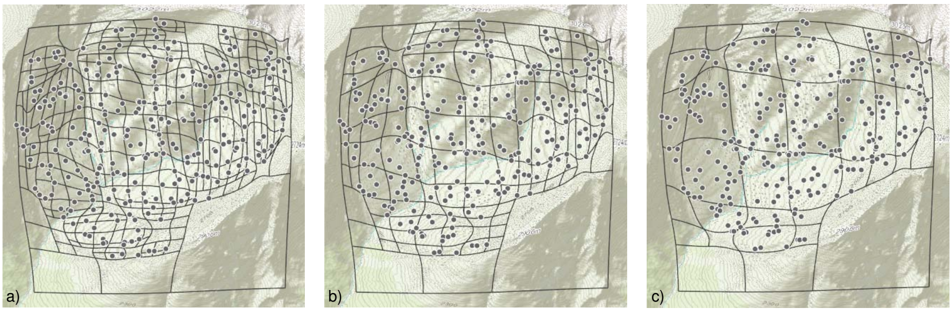


Figure 7.4.: Cartograms with different initial density grids at varying zoom levels: map zoom level a) 14 b) 13 c) 12. Base map: ©2013 ESRI ArcGIS Services

statistics (Table 7.1).

Considering the descriptive statistics for the three density grid zoom levels, the overall sum, average and maximum reduces between zoom levels 14 and 13, and slightly increases between zoom level 13 and 12. The standard deviation is gradually reduced, while the minimum length of the displacement vector increases. The distribution of the cumulative sum of the displacement vectors maintains its shape (Figure 7.5a-c).

Table 7.1 in the last two columns additionally compares the displacement of points at map zoom level 14 to the cartograms generated by a regular grid (see Figure 7.2a) and to KDE (see Figure 7.2b) as the initial point density grid. In both cases the overall displacement is significantly bigger, distorting the map significantly, especially in the case of the regular grid.

Meanwhile, the angular distribution of the mean (Figure 7.5d-f) levels off with decreasing zoom level for the depicted directions of the displacement vectors. That is, for a lower density grid zoom level the overall deformation is more uniform in map space.

Comparing the cartograms of the three density grid zoom levels shown in Figure 7.4, the overall deformation is smoother and the displacement more uniform, after a reduction of the grid zoom level. The change of the grid zoom level results in a smoother or more detailed spatial deformation depending on the selected zoom level. That is, the 'generalisation' level of the deformation is increased or reduced. This observation is in line with the increase or decrease of the cell size in a regular density grid (see Gastner and Newman (2004)) and the resulting spatial deformation.

7.2.3. Quadtree-based area cartograms

Basing the creation of the density grid on the quadtree allows the combination of the quadtree-based generalisation algorithms introduced in Chapter 6 with the cartogram, thus permitting the combination of space-directed and object-directed map generalisation. In principle, all discussed object-directed quadtree-based generalisation algorithms (see Chapter 6) can be applied. Figure 7.6 shows centrality-based simplification applied to a cartogram, with a quadtree-based density grid for map zoom levels 14, 13 and 12. The boundaries of the quadnodes are visualised to show the distortion of the map space for

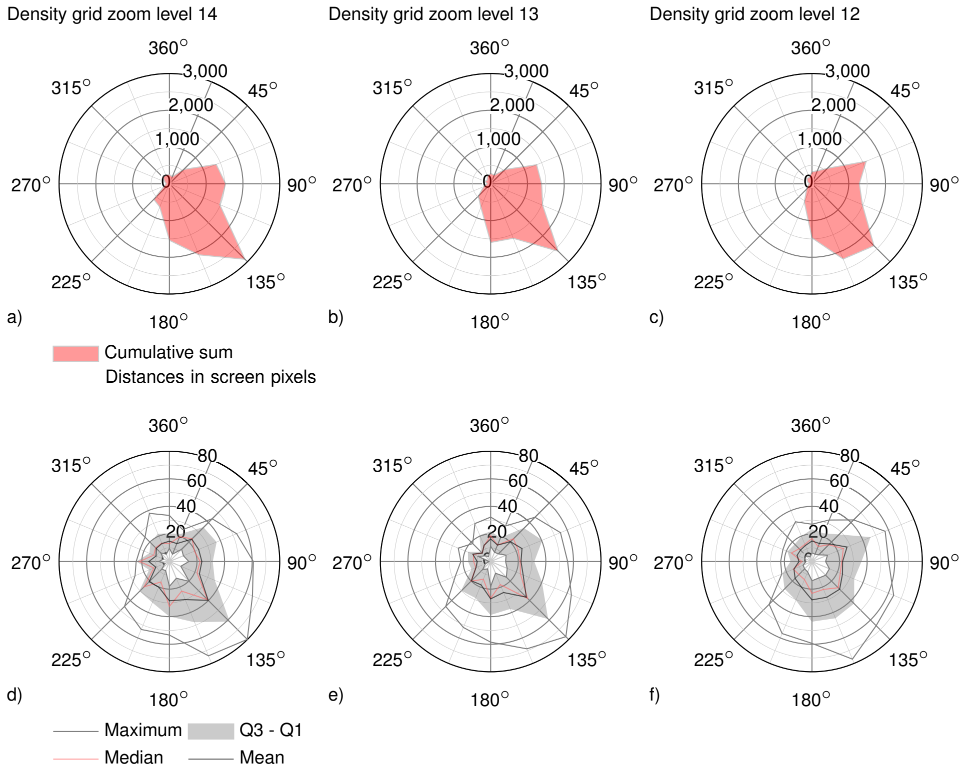


Figure 7.5.: Displacement vectors subdivided into 16 directional classes for the *cartogram* with different initial density grids for target zoom level 14 – 12 (a-c) denoting the cumulative sum in a-c, and the distribution of displacement vectors in d-f.

each map zoom level.

The map scale of the distorted map however, is not uniform throughout the map, as in the case of the undistorted map. Thus, on deformed regions on the map, the local map scale is either increased or decreased. That is, for a strongly distorted quadnode the object-directed generalisation algorithm may retain too many or too few map objects.

For each quadnode on a distorted map, the area of the quadnode either increases or decreases. If the area of the quadnode increases, more points than effectively retained by an object-directed generalisation algorithm may be retained. The inverse is true if the area of the quadnode decreases, or the shape is strongly elongated. In that case, less points than retained by an object-directed algortithm should be shown.

This effect can be observed in Figure 7.6 in the centre of the map, where too few points are kept, and in the transition area between transformed and untransformed areas where too many points are shown. This creates an artificial circular pattern around the deformed

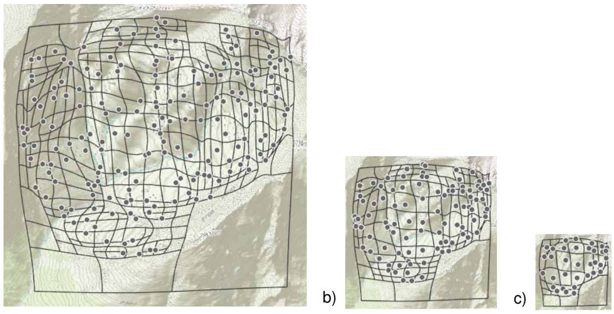


Figure 7.6.: Cartogram for map zoom level a) 14 b) 13 c) 12 in combination with centrality-based simplification. Base map: ©2013 ESRI ArcGIS Services

area.

Thus for distorted quadnodes with an increased area, a higher zoom level has to be used as a basis for object-directed generalisation, similar to the concept of *content zooming* in Chapter 4. The inverse holds for quadnodes with a reduced area or strongly elongated shape.

In addition to varying the map zoom level locally based on the quadnode shape and area, if full conflict resolution is required, quadnode border conflicts resolution can be applied as shown in Section 6.6. The quadnode border conflicts can be removed analogously to how they are resolved in the object-directed generalisation. However, if the shape of the quadnode is strongly deformed and the point symbol size exceeds the width of the quadnode, for border conflict resolution, second degree quadnode neighbours have to be considered.

7.3. Laplacian Smoothing

7.3.1. Working principle

Edwardes (2007) in his work investigates the use of continuous and deformable models of space for map generalisation, and whether the distortion of the map space itself could be used for map generalisation. Edwardes investigates Laplacian smoothing to reorganise points and relax spatial conflicts.

Laplacian smoothing was originally used in computer graphics to smooth the representation of surfaces. The implemented approach based on a primal-dual scheme Taubin (2001) proved sufficiently fast for interactive portrayal and was found suitable for contemporary mobile devices at the time. The work by Edwardes (2007) shows in detail the mathematical foundation and the adaptation to map generalisation. The implementation used in the following examples works in real-time, and is based on a modified implementation of Laplacian smoothing by Alistair Edwardes (Edwardes, 2007).

Like the cartogram in the previous part, Laplacian smoothing is also based on a grid as a means of deriving the distortion of the map. Thus Laplacian smoothing provides our second data-driven method as malleable space operator in space-directed generalisation.

a) b)

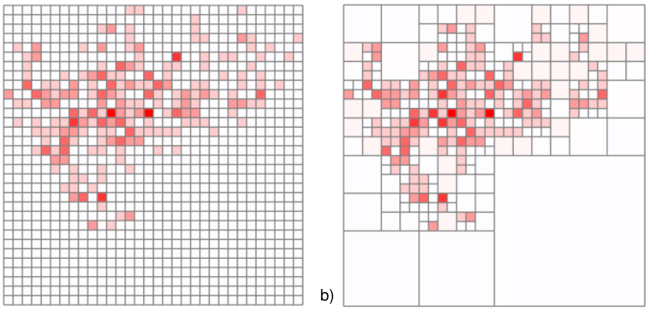


Figure 7.7.: Initial point density grids for Laplacian smoothing. a) regular grid with points per cell density b) quadtree derived grid with density estimation for each quadnode

a) b) c) d)

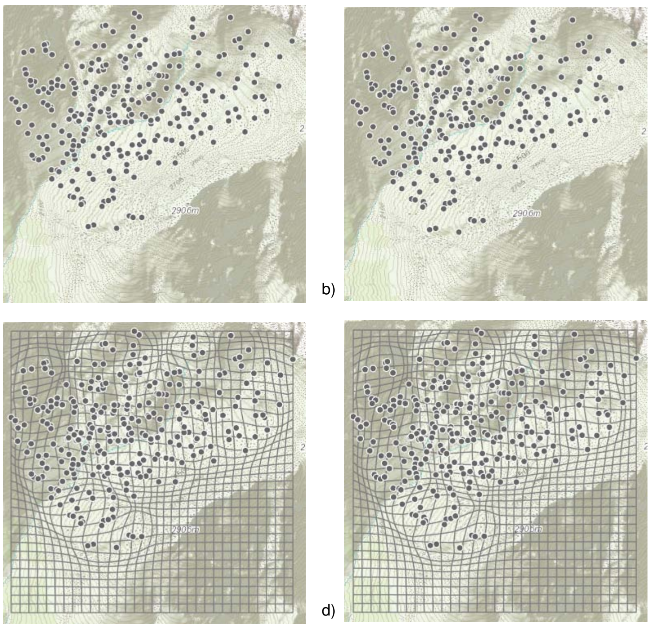


Figure 7.8.: Laplacian smoothing with two differently weighted grids: a) Regular weight grid (see Figure 7.7a) b) Quadtree grid with point density per quadnode (see Figure 7.7b). c,d) show the same maps as in a,b) but with the grid overlaid to better highlight the map distortion. Base map: ©2013 ESRI ArcGIS Services

Similarly to the cartogram-based method, the grid can be weighted and therefore parameterised by values derived from the quadtree. In order to facilitate the comparison of the two methods, the following examples use the same the map and dataset as used in the first part of this chapter (see Figure 7.1 and Appendix B).

Parameters of the Laplacian smoothing are, the *number of grid cells* (and therefore the grid cell size), the *weight* attributed to each grid cell, and a *smoothing factor*. The cell size influences the area that each point that is added to the grid influences. The weight then affects how much the cell distorts the map space.

The higher the weight parameter the stronger the distortion. Among the parameters, weight has the strongest impact on the deformation of the map. The smoothing factor is a nine cell kernel that applies spatial averaging to the neighbouring cells of the weighted cell, comparable to low-pass filtering. It expands the area of influence of the weight set, inside a single cell, to its neighbouring cells and therefore further emphasizes the weighted cells.

Figure 7.7a) shows the regular grid weighted by the number of points per grid cell, whereas b) depicts the grid based on the quadtree, weighted by the number of points per quadnode area. In the presented maps the weights are empirically set: weight $w = 10$ and smoothing factor $sf = 2$. Equation 7.1 shows how the weights of a grid cell in Laplacian smoothing are obtained. Both grids for Laplacian smoothing and the diffusion-based cartogram are 64 x 64 wide, and are derived from the same dataset.

$$w_{cell} = w\frac{p_{node}}{a_{node}}\frac{a}{n_{grid}}. \tag{7.1}$$

w_{cell}	Grid cell weight
w	Weight factor
p_{node}	Number of points stored in a quadnode
a_{node}	Quadnode area
a	Quadtree area
n_{grid}	Number of grid cells

The results of Laplacian smoothing for both grids are shown in Figure 7.8. Weights and smoothing factors were set empirically. Compared to the cartogram, the weighting of the grid is less prone to strong, unwanted distortion.

Subfigures a) without and c) with the distorted grid overlaid, represent Laplacian smoothing derived from the regular grid shown in Figure 7.7a). Subfigures b) and d) denote Laplacian smoothing based on the grid with the weights derived from the quadtree.

Overall both results show only slight differences. The average length and the standard deviation of the displacement vectors only differ slightly, in the range of less than a tenth of a screen pixel.

Both maps increase the cell size in the regions of high point density and show strong deformation towards the border of the point pattern. The overall area of deformation does not extend significantly over the boundary of the point pattern. Thus, for points located in the centre of the point pattern, their cells have only limited space to extend. This implies that the overall shape of the point pattern is more or less maintained.

The similarity of the distortion in both cases provides the possibility of using a quadtree-based weighting scheme to control space-directed generalisation, and to use Laplacian smoothing in combination with quadtree-based object-directed generalisation algorithms. Additionally cell size, weighting and smoothing factor can be linked to the zoom level.

7.3.2. Laplacian smoothing – zoom dependent effects of the density grid

Figure 7.9 shows Laplacian smoothing based on different quadtree-based weighted grids for zoom levels 14 – 12, keeping the gridsize, weight and smoothing factor fixed. With the reduction of the zoom level the deformation of the map space is smoother and more uniform across the map space.

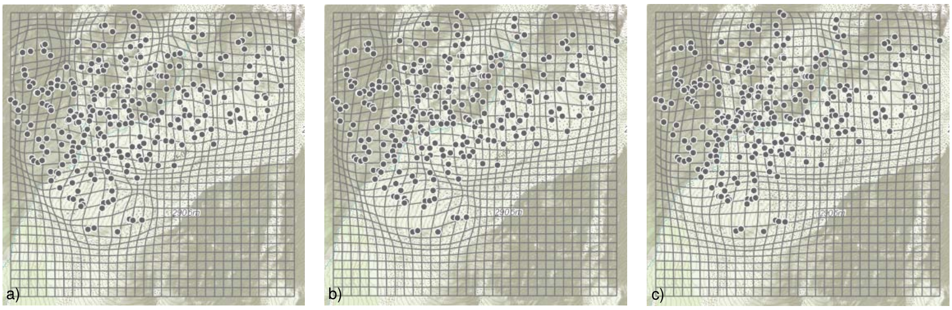


Figure 7.9.: Laplacian smoothing with different initial density grids at varying target zoom levels: map zoom level a) 14 b) 13 c) 12. Base map: ©2013 ESRI ArcGIS Services

The descriptive statistics for the length of the displacement vectors for all points in each of the three maps in Figure 7.9 shows for all values, except for the minimum length of the displacement vectors, increasing values. The evolution of the length of displacement vectors from zoom level 14 to 12, confirms alongside with the reduction in the standard deviation, that the deformation of the map space becomes smoother. The reduction of the mean shows that the deformation of the map overall decreases.

Table 7.2.: Descriptive statistics for Laplacian smoothing displacement vectors with different quadtree density grids

Map Zoom	14	13	12
Min	0.26	0.30	0.42
Max	21.91	19.33	18.25
Mean	9.57	9.22	8.28
Standard deviation	4.29	4.32	3.94
Sum	2718.73	2620.68	2352.53

The angular distribution of the cumulative sum into 16 directional classes indicates that the directional displacement is maintained (Figure 7.10a-c). A similar, but less distinctive, picture shows up in the descriptive statistics for each directional class (Figure 7.10d-f).

By changing weights in the grid for Laplacian smoothing, based on different quadtree-based zoom levels, the overall deformation of the map space does not change significantly, but actually decays. Thus, an increase in deformation by changing the zoom level, for instance by zooming into the map, can be controlled by changing the grid size and/or changing the weights.

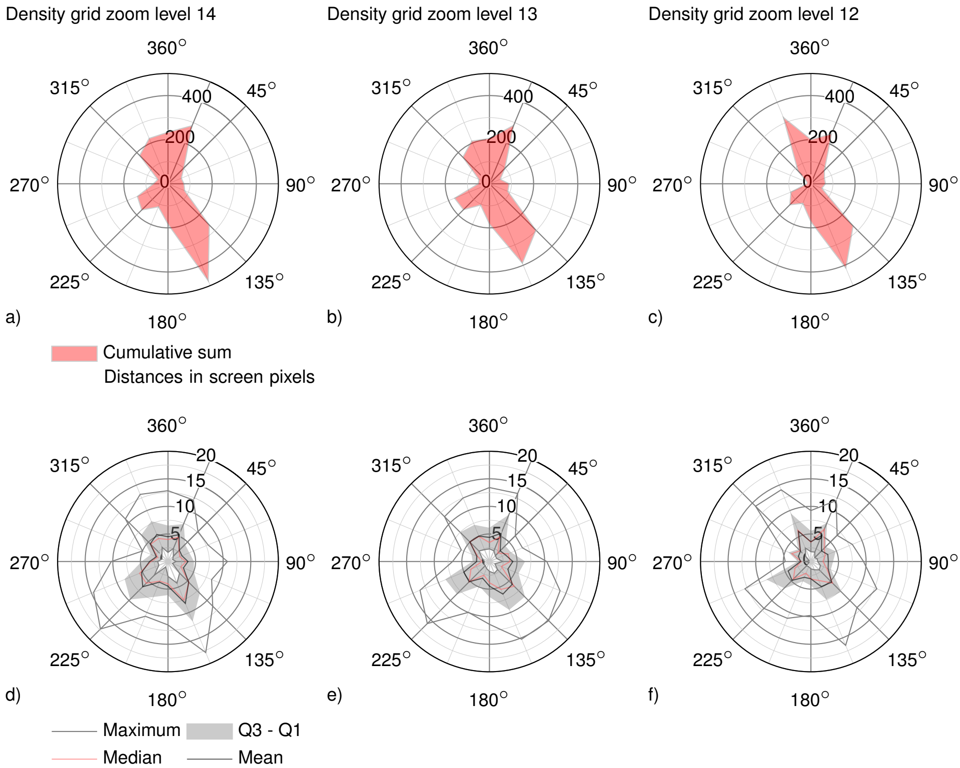


Figure 7.10.: Displacement vectors subdivided into 16 directional classes for the *Laplacian smoothing* with different initial quadtree density grids for different target zoom levels 14 – 12 (a-c) showing the cumulative sum in a-c, and the distribution of displacement vectors in d-f.

7.3.3. Quadtree-based Laplacian smoothing

The previous section showed that changing solely the weights of the distribution based on the quadtree equalizes the spatial deformation but does not drastically change the degree of deformation. The change of the weights w or the gridsize provides the possibility of combining Laplacian smoothing as a space-directed generalisation operation, in conjuc-tion with object-directed quadtree-based generalisation algorithms.

The generalisation sequence shown in Figure 7.11 shows Laplacian smoothing for map zoom levels 14, 13 and 12 in combination with centrality-based simplification. The grid-size used for Laplacian smoothing is equal to the smallest quadnode size of the applied map zoom level. That is, the number of grid cells is equal to the maximum possible number of quadnodes per zoom level.

The generalisation sequence, based on the quadtree, keeps the size of the grid cells constant all three zoom levels[2]. Thus in Figure 7.11a) the number of cells are 32 x 32, for map b) 16 x 16 and 8 x 8 in map c). Equally, the value of the weight is kept constant ($w = 10$) for the three zoom levels.

The distortion of dense regions and the resulting variation of the local map scale result in smaller and larger quadnode areas than on the undistorted map. The object-directed generalisation algorithm operates on the basis of the non-deformed map area. In the same way as in the cartogram-based deformation (see: Section 7.2.3) more or less points are retained, as the map scale of the quadnode increases or decreases, respectively.

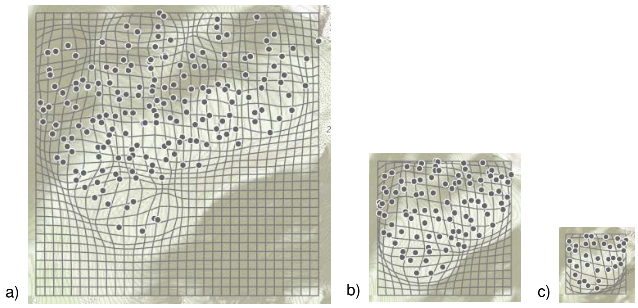


Figure 7.11.: Laplacian smoothing for map zoom level a) 14 b) 13 c) 12 in combination with centrality-based simplification. Base map: ©2013 ESRI ArcGIS Services

For map zoom level 14 (Figure 7.11a) the overall map deformation is moderate compared to map zoom levels 13 or 12. Hence, the generalisation result does not show regions where more or less points than expected are retained. It is not solely the moderate deformation at map zoom level 14 supporting this impression, but also the fact that Laplacian smoothing largely maintains the shape of the weighted grid cells.

The shape of cells situated around the weighted cells is more affected by deformation. Most affected are those cells that lie just outside of the point distribution (see the South West area of the point pattern in Figure 7.11a). Neighbouring weighted cells seem to even out the distortion, such that the shape is better maintained.

The previous Section 7.3.2 showed that the change of the quadtree-based grids – while keeping the number of grid cell constant – hardly affects the amount of the map deformation, and mainly influences the smoothness of the deformation. The reduction in the number of grid cells at map zoom levels 13 and 12 (Figure 7.11b,c) goes with a smoother but stronger deformation of the map space. Compared to those maps with constant number of grid cells, Figure 7.11b and c show that the reduction in the number of grid cells increases the deformation of the map.

Hence, by increasing the deformation and applying the object-directed generalisation operators, the mismatch between retained points and available map space increases. Similarly, but to a lesser extent than in the cartogram-based example of in Figure 7.6c, Figure

[2] A similarly strong deformation while maintaining the number of cells, can be achieved by a 'strong' increase of the weight factor. This result, however, in a rather 'angled' deformation, as many neighbouring cells share the same values due to the comparatively small cell size in combination with quadtree-based weighting.

7.11c shows an accumulation of points at the border of the point pattern for map zoom level 12. That is, at the border of the distribution of this point pattern more points are retained than desired, while in the strongly deformed centre less points are kept. Thus, the density of the point pattern is poorly retained if the combination of space- and object-directed generalisation does not better account for the changes in the local map scales after the deformation.

As noted in Section 7.3.2, for distorted quadnodes with an increased area the points from the childnodes can be retained, leading to a locally higher map zoom level. For quadnodes with a reduced area or a strongly deformed shape, the inverse process applies. For full conflict resolution of overlaps on quadnode cell borders, border conflict resolution can be applied by either removing or moving points if they are conflicting.

7.4. Discussion and concluding remarks

Space-directed generalisation, or the concept of a malleable space, enables the generalisation and transformation of background features of the map, together with its foreground features. The two case studies presented in this chapter – the animal observations in the Swiss National Park and the distribution of restaurants in the city of Zurich introduced below – provide a means of comparison between the two methods that implement the concept of the *malleable space* as a part of space-directed generalisation. An overview of the differences between cartogram-based transformation and Laplacian smoothing is given in Table 7.3.

In addition to the dataset of animal observations in the Swiss National Park, this section presents a second dataset (see Figure 7.12), featuring a different point pattern with several distinct clusters (see KDE in Figure 7.12d). The dataset represents POI data; here, restaurants in the city of Zurich. Map (a) presents the cartogram and map (b) the Laplacian smoothing. For both maps the quadtree shown in subfigure c is applied.

The two methods have in common that they distort the map background together with the overlaid foreground features. Both methods are data-driven, are parameterisable and can be executed in real-time. In addition they allow for a parameterisation in combination with the quadtree, and further, allow for a combination with object-directed generalisation algorithms.

Due to the different working principle of the methods, the parameterisation affects the deformation in different ways. While in the case of the cartogram the quadtree serves as a binning structure with attributed weights, in the case of Laplacian smoothing it serves as a means of applying weights to the grid structure of the method.

Comparison between Cartogram and Laplacian Smoothing In order to preserve the readability of the diffusion-based map cartogram, it depends highly on an accurate density equalisation of the point distribution. That is, the points as discrete objects must be converted and binned into a continuous surface grid. Therefore an acceptable balance between the bin sizes and density needs to be found, in order to prevent substantial distortions in regions of high densities. The selected parameterisation based on the quadtree principle provides such a balance and maintains the readability of the map. Unlike with

Table 7.3.: Comparison between Cartogram and Laplacian smoothing

	Cartogram Gastner and Newman (2004)	**Laplacian Smoothing** Edwardes (2007); Taubin (2001)
Domain	Particle physics	Computer Graphics
Principle	Density equalisation	Laplacian Smoothing
Performance	Real time computation depends on the number of grid cells	Real time computation depends on the number of grid cells
Function	Density function	Weighting function
Factor	Cell size, density, density variation	Cell size, weights
Global parameter		Weight and smoothing factor
Bound	Implementation includes a boundary region, deformation exceeds original bounding box	Deformation is zero at the bounding box of the distribution (implementation dependent)
Deformation	Strong local deformation possible	Local moderate weight dependent deformation
Shape preservation	Preservation depends on density distribution, including strong local distortion and an overall extension of the point pattern	Good maintenance of the overall spatial distribution of the point pattern
Directionality	Directionality is present due to the density equalisation principle	No apparent directionality of the displacement vectors
Singularity	Less susceptible to strong cell shrinkage	Strong shrinking of grid cells to one point possible
Generalisation	Quadnode size based on map zoom level	Quadnode size based on map zoom level
Parameterisation	Difficult and dependent on density distribution	Easier and less prone to unwanted strong local deformation

Laplacian smoothing, the bin size, the attributed density to the area of the bin and the variation between neighbouring bin cells significantly affects the distortion of the map.

Laplacian smoothing then, is less prone to strong unwanted distortions, as the magnitude of the distortion depends on the bin size and global weight settings. In comparison to the cartogram, Laplacian smoothing provides two additional global parameters, a weighting and a smoothing factor, to control the spatial deformation. However, the magnitude of the distortion is generally less pronounced than it is in the case of the cartogram.

In both methods the deformation of the map space is zero at the boundary of the map. Due to computational reasons, the implementation of the cartogram sets the boundary of the map bounding box slightly larger than the bounding box of the cartogram (see Figure 7.12a, b).

The map used in the previous sections exhibits a point pattern with most points located in the North-Western part of the map and a nearly empty South-East corner (see Figure

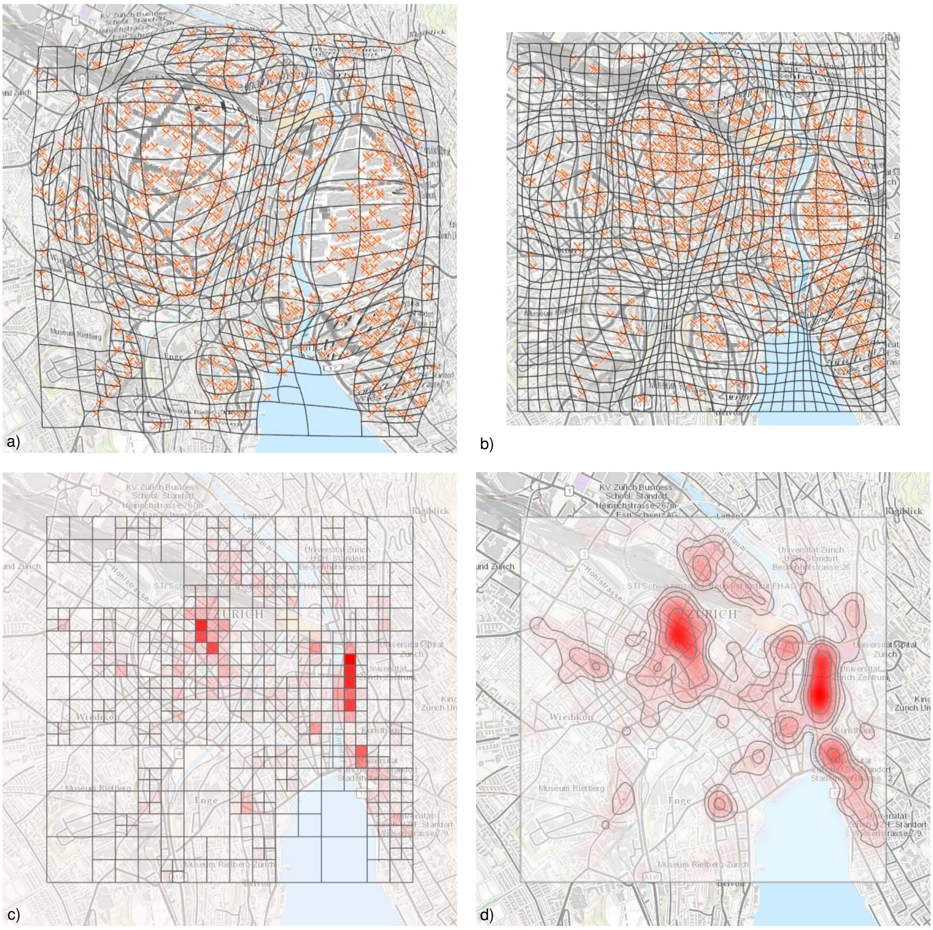


Figure 7.12.: Distribution of restaurants in the city of Zurich at map zoom level 14 a) Cartogram b) Laplacian Smoothing c) Density grid used for both maps d) KDE of the point pattern. Base map: ©2013 ESRI ArcGIS Services

7.2b). A comparison of the displacement vectors between the Cartogram algorithm (Figure 7.5) and Laplacian smoothing (Figure 7.10) shows that the directions of displacement differ significantly.

On the cartogram the points are mainly displaced to the South-East, due to the underlying density equalisation principle. That is, the points are preferably displaced towards the empty areas of the map. The displacement vectors of Laplacian smoothing, however, mainly point to North and South-East. That is, these points are displaced into directions where less points are present. The chosen dataset nicely highlights these differences as a consequence of the different working principle of the algorithms. These are, however, less pronounced in other datasets (Figure 7.12).

Qualitatively, Laplacian smoothing preserves the shape of the point pattern better, generally, than the cartogram, especially in dense areas of the point distribution. This is on the one hand due to the overall different degree of deformation, which in the two use cases presented is higher for the cartogram. On the other hand it is also due to the different working principles of the algorithms. That is, Laplacian smoothing does not equalize regions of high densities and keeps the deformation local, which results in a better shape preservation.

However, a downside of the 'local' nature of Laplacian smoothing is that, as opposed to the cartogram algorithm, stronger shrinking of cells may occur, creating 'singularities'. See, for instance, in Figure 7.12b along the Northern lake front, where in that region some grid cells are shrunk down nearly to a point. Such 'singularities' may occur in Laplacian smoothing for cells between strongly weighted regions, if the global weights are set to high values.

Combination with quadtree-based object-directed generalisation algorithms The previous sections presented methods of a data-driven malleable space in combination with a quadtree-based object-directed generalisation algorithm. That is, based on the same methods as presented in Chapter 6, object-directed generalisation methods can be applied. These methods derive the degree of generalisation from the current map zoom level. in space-directed generalisation, however, the zoom level is not uniform throughout a map.

As a consequence, if the local zoom level on the distorted map increases, the represented area is increased and therefore more space is available and spatial conflicts are reduced. This means that the generalisation algorithms retain less points than expected, reducing the perceived point density of the point pattern. The opposite is true for locally reduced zoom levels and especially for highly distorted regions, creating further clutter and overlap and therefore an increase in the perceived density. This introduces an artificial circular pattern around the border of the significantly distorted areas.

Thus, in order to combine object- and space-directed approaches, the object-directed generalisation algorithms need to consider the local distortion of the map scale. That is, with quadtree-based algorithms the local distortion should be investigated per quadnode. Hence, for each quadnode at a given zoom level, the algorithm considers its distortion, by comparing the ratio of the diagonal lengths and the quadnode area. A ratio much smaller or higher than one between the two diagonals of a quadnode hints towards a strong distortion of a quadnode. A quadnode with strong distortions or reduced area then reduces the local content zoom level, while in the case of an increased node area the local content zoom level is increased.

In addition to including the local distortion of the map scale, border conflicts may be removed in the same way as in the undistorted case (Section 6.6). The algorithm, however, needs to consider the transformed position in order to remove border conflicts. It may happen that with strongly distorted quadnodes second- or third-degree neighbouring quadnodes have to be included to remove possible border conflict.

Outlook This chapter illustrated how the concept of a malleable space based on two different algorithms, the Cartogram implementation by Gastner and Newman (2004) and Laplacian smoothing by Edwardes (2007), can be applied as a part of space-directed generalisation and incorporated into the overall methodology of point generalisation (see Chapter 3).

Further steps to include are the application of higher resolution background maps for strongly enlarged map regions, and the inclusion of attribute weighted densities to enlarge regions based on values, rather than merely basing this on the quantity of points. Furthermore it is possible to integrate Laplacian smoothing directly into the quadtree-based generalisation.

In addition, as introduced by Edwardes (2007), Laplacian smoothing allows the formulation of constraints and the application of these to regions, such that these regions are safeguarded and not affected by distortions (e.g. linear features).

Finding a satisfactory parameterisation of malleable space algorithms is, even with the help of a second data-driven index structure, not quite straightforward, and difficult to balance over a range of scales and thus requires further empirical investigation and fine-tuning. Also, for both space-directed transformations, the perceptual and cognitive validity remains to be evaluated (Carpendale, 2001; Mountjoy, 2001; Lam et al., 2007; Schafer and Bowman, 2003; Pietriga et al., 2010).

Chapter 8.

Experiments and Results

The previous chapters discussed the development of real-time map generalisation algorithms that enable a flexible and modular generalisation process. These real-time algorithms provide solutions for both object-directed generalisation operators, described in Chapter 6 and space-directed generalisation, described in Chapter 7, respectively. The two algorithms available for space-directed generalisation – density-equalising cartograms and Laplacian smoothing – have already been demonstrated and briefly evaluated in Chapter 7. Most of the algorithms proposed in this work, however, follow the object-directed paradigm of generalisation. While Chapter 6 has brought an initial illustration of the use of some these algorithms, a thorough analysis of their computational and cartographic performance is still remaining. This chapter, therefore, provides further examples of the use of object-directed algorithms, and conducts an in-depth, quantitative cartographic analysis based on three case studies, using the analytical toolbox described in Chapter 5. This chapter provides a first analysis of the results, while the in-depth discussion of these results is given in Chapter 10.

8.1. Dataset and Prototype System

The datasets used for the cartographic analysis of the object-directed generalisation algorithms in this chapter are 1) the SwissLichens dataset, 2) the animal observations of the Swiss National Park, and 3) POI data from the region of Zurich. The different volumes and characteristics of the datasets serve to systematically assess the algorithms and perform the cartographic analysis. Appendix B provides the details of these and other datasets that were used in the course of this thesis.

The proposed algorithms perform in real-time on a prototype map client (see Appendix A for details). The implementation is based on Java and Processing (www.processing.org). In addition to the point generalisation module and the support of selected map services, the development environment contains a module for the multiscale cartographic analysis of the resulting generalised point data sets (Chapter 5) and the content zooming module (Chapter 4).

Based on this prototype system, Chapters 6 and 7 have shown examples of how the different operators for object-directed and space-directed generalisation, respectively, can be implemented, and how the concept of content zooming is applied (Chapter 4). All figures in this thesis, apart from the diagrams, are screenshots taken directly from the prototype system.

8.2. Performance

One of the key performance criteria of an algorithm is certainly its speed of a particular algorithm, necessary to achieve real-time behaviour. While this might seem a particularly *measurable* criterion, it turns out to be rather tricky, as for practical purposes, the speed that can be obtained does depend largely on the particular set-up and the technology that is used in the implementation. If the generalisation is computed on the server-side, then obviously much more complex computations are possible in a tolerable response time than if generalisation takes place on a mobile client. Thus, the criterion of computational speed in our case is restricted to the client side. The performance analysis is conducted on a desktop dual-core Pentium CPU running at 3.16 GHz, with Windows 7 and Java 1.6. Although this is not a mobile client, the performance characteristics of this testing platform are in a similar order of magnitude.

A performance test of the computation time used for quadtree creation as well as for selected object-directed generalisation algorithms is shown in Table 8.1, for three datasets of increasing volume. The computation time further depends on the spatial distribution of the dataset, the implementation, and the computing architecture used. The measure distinguishes between the average creation time of the quadtree for each dataset and the average time needed to update between the individual zoom levels, with no caching option applied. While there is no significant difference between the selection and aggregation algorithms, displacement with its neighbourhood check is (not surprisingly) more costly.

As the tests reported in Table 8.1 demonstrate, real-time performance can be achieved with the proposed algorithms, with response times that are clearly in the sub-second range even for large datasets. For very large datasets the caching option further reduces the update time between the zoom levels, and provides a smoother zooming experience to the user. It also provides a significant reduction of computation times between zoom levels as it only has to be performed once per zoom level and current spatial extent.

Table 8.1.: Average execution time for three different datasets, for quadree creation and update time between two consecutive different zoom levels, on a dual-core Pentium CPU running at 3.16 GHz, with Windows 7 and Java 1.6.

	Small data set Animal observation SNP ~290 points	Medium data set POI restaurants Zurich ~2800 points	Large data set SwissLichens ~86000 points
Quadtree creation	0.8 ms	8 ms	460 ms
Average computation time to move between two different zoom levels			
Selection	0.13 ms	3 ms	150 ms
Simplification	0.06 ms	2 ms	110 ms
Aggregation	0.05 ms	1.6 ms	90 ms
Displacement	0.98 ms	13 ms	800 ms

8.3. Cartographic Analysis

Within the literature on automated map generalisation, the importance of the assessment of generalisation, and the need for further research has often been advocated (Weibel and Dutton, 1999; Bard, 2004b; Mackaness and Ruas, 2007; Joao, 1995). Overall map evaluation is regarded as a rather complex task and has been addressed by various researchers and research projects (Bard, 2004a; Joao, 1995; Harrie and Stigmar, 2010; Stigmar and Harrie, 2011; Ehrliholzer, 1995; McMaster and Shea, 1992; AGENT, 1998).

In evaluating map generalisation, Mackaness and Ruas (2007); Weibel (1995) distinguished three different stages of evaluation. *Evaluation for tuning* **prior** to the generalisation process, triggering generalisation and its parametrisation; *evaluation for controlling* **during** the generalisation process, controlling it; and finally *evaluation for assessing* **after** the process to evaluate the outcome and the overall quality of the solution.

Evaluation for assessing the generalisation is further subdivided into *evaluation for editing* aiming to identify errors, a *descriptive evaluation* of how the map content has changed, and *evaluation for grading*, which tries to quantify the quality of the generalisation result with a grade.

In addition to the complexity stated by Mackaness and Ruas (2007), an evaluation framework should be able to handle the notion that the final output is a compromise among a set of sometimes competing map objectives, and that therefore satisfactory generalisation results can be reached using different generalisation processes and strategies.

The cartographic analysis presented in this chapter falls into the last category and concentrates on the identification of errors and a descriptive evaluation of the generalised dataset. The analysis, however, does not aim to quantify the overall quality of the results, as they depend highly on the given map purpose and the scale of the phenomena and is therefore not generic. The evaluation concentrates rather on giving insight into the developed solutions and the strengths and weaknesses of the object-directed algorithms described in Chapter 6. The cartographic analysis is driven by the following set of questions leading through seven different cartographic aspects investigated.

Data reduction: What is the reduction rate in relation to the zoom level? How does the reduction rate compare to the Radical Law by Töpfer and Pillewizer (1966)?

Cartographic conflict reduction: How many overlaps between point symbols are resolved? Are legibility constraints maintained?

Data enhancement: How are important point attributes retained? How does the spatial distribution compare to the source distribution?

Displacement measures: How is displacement achieved locally and globally? How do the displacement vectors differ between the generalisation algorithms?

Maintenance of spatial patterns: To what degree is the density maintained over the range of scales? Where locally does the density change most and is the overall point distribution maintained?

Homogenisation Is there a tendency towards a more uniform spatial distribution? Does the spatial distribution of the generalisation outcome show homogeneous, regular patterns?

Cluster maintenance Are the clusters maintained over the range of scales? And if the clusters vary in form and size, where do they change?

8.4. Data Reduction

This section analyses the data reduction rate of the object-directed generalisation algorithms introduced in Chapter 6. The data reduction rate provides a metric of the degree to which the generalisation algorithm reduces the amount of content on the thematic foreground layer. And how does the reduction rate compare to the well-known *Radical Law* by Töpfer and Pillewizer (1966)?

Based on the Swiss Lichens Database, Figure 8.1 shows the data reduction curves for the quadtree-based algorithms, as well as for the Radical Law shown in Equation 8.1. Not surprisingly, point reduction algorithms (selection, simplification, aggregation) retain less points than displacement, for those zoom levels where most conflicts arise. It also becomes clearly visible that the quadtree-based algorithms retain more points at higher zoom levels than the Radical Law would suggest (which depends on the scale denominator of the source map or base zoom level; see Figure 8.2).

$$n_f = n_a \sqrt{\frac{M_a}{M_f}} \qquad (8.1)$$

n_f	Number of derived objects
n_a	Number of original objects
M_a	Scale denominator of source map (base zoom level)
M_f	Scale denominator of derived map (target zoom level)

This dramatically different behaviour may be surprising, but is due to different underlying principles of point reduction used. The Radical Law was originally empirically developed using topographic map series; hence it does indeed reflect a good first estimate of the objects to be retained. An extended version of the Radical Law additionally considers the symbol sizes (see modified Radical Law by Burghardt and Cecconi, 2007). However, when used as a measure to guide the generalisation process, the spatial arrangement of point features must be controlled by the generalisation algorithm applied, since the Radical Law merely uses the ratio of the square roots of scale denominators and thus has no concept of space *per se*.

The quadtree-based generalisation algorithms, on the other hand, inherently take the proximity and density of point symbols into account, owing to the quadtree structure. The data reduction curve depends highly on the number of points and the nearest neighbour distribution among the point symbols. Quadtree-based algorithms resolve cartographic conflicts at all zoom levels, but do not follow the empirically derived Radical Law.

These algorithms will only remove or aggregate points if spatial conflicts occur. Thus, between levels 20 and 15 in Figure 8.1 almost no point reduction takes place. Once the average distance between data points reaches the size of the quad cell corresponding to the given zoom level, however, the point reduction rate rapidly increases. At this point the slope of the data reduction curves becomes even steeper than the one of the Radical Law.

In special cases however, generalisation algorithms guided by the Radical Law may not solve all the cartographic conflicts at certain zoom levels. This is the case for highly clustered datasets, if the number of cartographic conflicts far exceeds the number of points to be retained by the Radical Law.

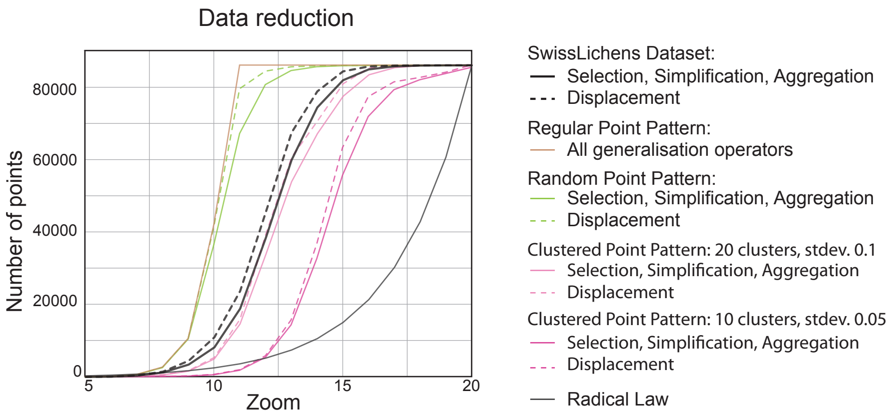


Figure 8.1.: Data reduction for different datasets with the same spatial extent and number of points; SwissLichens, as well as artificial regular, random and clustered point patterns

In Figure 8.1 the data reduction of the generalisation operators applied to the SwissLichens dataset is also presented in context with three different, artificially generated, datasets, with the same number of points, and applied to the same spatial extent (given by the axis-aligned bounding box).

The more regular the spatial distribution, the steeper the S-shape of the curves. A similar, but less distinct, curve progression is shown by the random point pattern, as it does not contain distinct clusters that are reduced by the generalisation operators at higher zoom levels.

Finally, two samples of clustered point patterns are also plotted in Figure 8.1. The generated patterns differ in the number of clusters, their size and the standard deviation around the cluster centre. Depending on their parameterisation, the S-curve of the data reduction shifts its inflection point to higher or lower zoom levels,respectively.

For all datasets used, quadtree-based algorithms retain and reflect the underlying spatial configuration of the respective point dataset in the data reduction curve. They retain more points at higher zoom levels and show a sharper decrease after the inflection point than if the Radical Law would have been applied, even if the base zoom level of the Radical Law is shifted, as indicated in Figure 8.2.

Topographic datasets typically have a particular map scale, and are cartographically correctly rendered with no overlaps, while thematic datasets do not necessarily have a clearly defined map scale (base zoom level). That is, the features in a thematic dataset are eventually small – such as in the lichens data set, or coincide, such as services icons in POI datasets (toilets, restaurants, shops), or contain overlaps – such as in datasets of

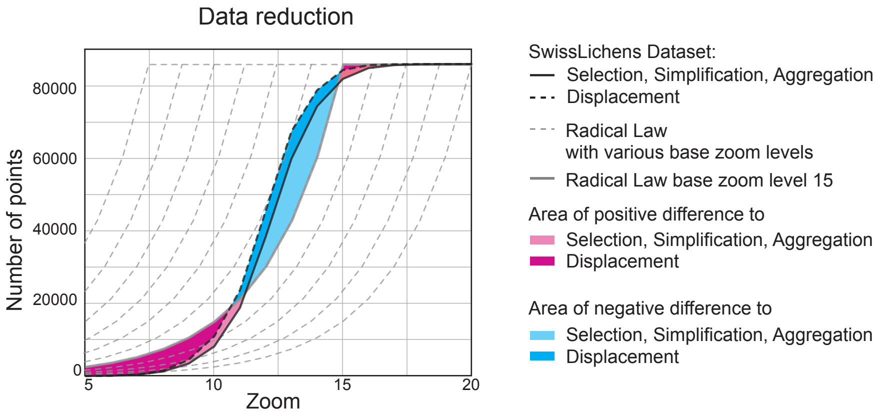


Figure 8.2.: Data reduction for the SwissLichens dataset, with quadtree-based reduction and displacement algorithms compared to the Radical Law with different base zoom levels (or M_a Scale denominator of source map (cf. Equation 8.1). Areas of positive differences are marked in pink and denote a superior number of points for algorithms following the Radical Law compared to the quadtree-based algorithms, while areas of negative differences are shown in blue and denote an inferior number of points.

pictures taken from the same location.

Thus, even if the base zoom level is shifted, as shown in Figure 8.2, to minimize cross-section between both curves, the different nature of the curves show areas of positive differences in very high and low zoom levels, meaning that the Radical Law retains more points than the quadtree-based algorithms, and an area of negative difference where the Radical Law retains comparatively less point data than the quadtree-based algorithms.

Figure 8.3 qualitative impression of the quadtree-based algorithm (a), as opposed to a typification process (Burghardt and Cecconi, 2007) guided by the Radical Law (b) and a global selection (c), based on an artificial clustered dataset with 200 points. The base zoom level is set such that for the typification process at map zoom level 9 no overlap occurs between the point data. With this configuration, compared to the quadtree-based algorithm, the cartographic result seems more cluttered at lower zoom levels and less cluttered at higher zoom levels (similar to what is shown in Figure 8.2, altough with a different data set).

Including the space occupied by map symbols for the same region, the ratio between the area covered by map symbols (non-white pixels) and the map area increases, as shown in Table 8.2. The ratio has been defined based on a raster overlay (Figure 8.4b) and the map screen area (Figure 8.4a) with reference map zoom level 11 (map scale ~1:288’895). The raster subdivides the map area into cell sizes equal to the symbol size, which for each consecutive zoom level doubles. A pixel is marked as occupied if the map point falls within a raster cell (Figure 8.4b). In Figure 8.4a the area each point symbol covers is represented by a rectangle symbol at the point position. In both subfigures, occupied

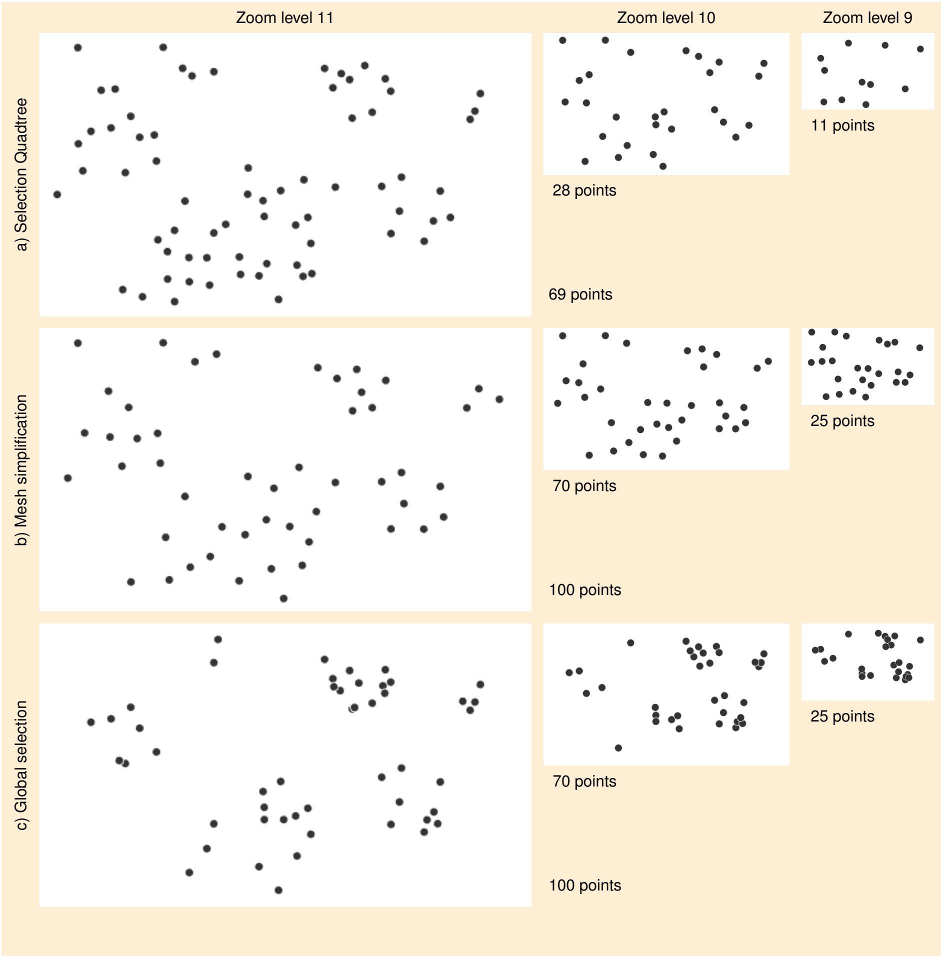


Figure 8.3.: Data reduction for a) quadtree-based selection algorithms; b) typification based on mesh simplification (Burghardt and Cecconi, 2007) c) and global selection (without considering cartographic conflicts). In b) and c) the data reduction is guided by the Radical Law.

pixels for the consecutive zoom levels range from dark red, at zoom level 11, to light red, at zoom level 8. In both cases – map screen ratio and raster ratio – the ratio increases non-linearly with decreasing map zoom, similarly to the data reduction curve (see Figure 8.1).

Table 8.2.: The ratio of 'black' pixels in relation to the map screen area for different zoom levels, based on map screen pixels or on a raster overlay. The 'black' pixels are derived from the map output of quadtree-based selection applied to the SwissLichens dataset for the Canton of Grisons, Switzerland. See Figure 8.4

Zoom level	11	10	9	8
Map screen ratio	0.044	0.069	0.098	0.153
Raster ratio	0.065	0.129	0.155	0.34

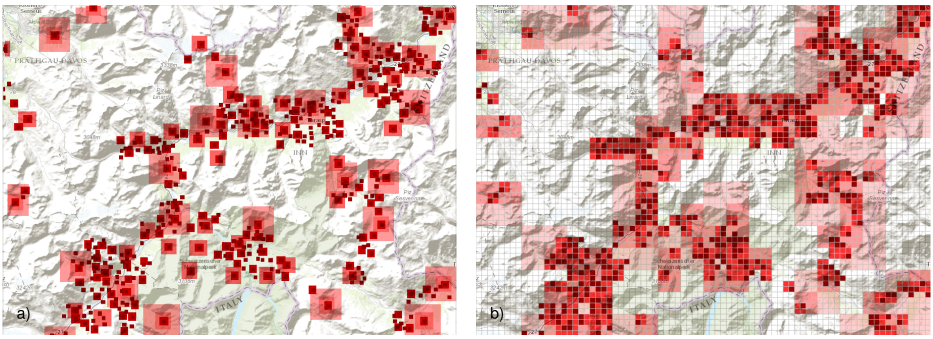


Figure 8.4.: 'Black' pixels for consecutive map zoom levels 11 (dark red) to 8 (light red): a) on the screen area; b) on a raster overlay. Base map: ©2013 ESRI ArcGIS Services World Topo Map

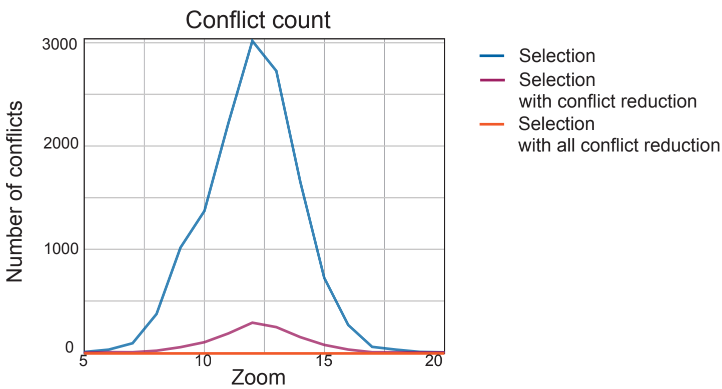


Figure 8.5.: Conflict counts for selection without conflict checks, and selection with different forms of conflict checks applied. Number of points in SwissLichens: 86,845.

8.5. Cartographic conflict reduction

Zoom levels with a high data reduction rate in Figure 8.1 hint to a high number of cartographic conflicts – that is, overlapping symbols – at those zoom levels. Figure 8.6 shows the evolution and reduction of cartographic conflicts over the different zoom levels for the selection algorithm, with three different variants of conflict resolutions applied. The point reduction algorithms (selection, simplification, aggregation) only retain a single point per quadnode for the target zoom level (see Chapter 6). However, cartographic conflicts are not entirely removed by solely retaining one point per quadnode, as this does not consider potential overlaps of points contained in neighbouring quadnodes (Figure 8.5, blue curve). That is, two point symbols might be lying across the border of two neighbouring quadnodes, separated by a distance less than the symbol size. This can be alleviated by checking for collisions in the four – respectively eight – possible quadnode neighbours (Figure 8.5, red line), or by moving all points towards the centre of the quadnode, not allowing for any overlap (Figure 8.5, orange line). The application of conflict constraints is described in Section 6.6.

Figure 8.6 shows results of applying the different variants of conflict constraints, in the case of value-based selection of lichens data for the SW part of Switzerland – the lower part of the Swiss Rhone valley. While pure selection still shows a considerable number of cartographic conflicts (red dots in Figure 8.6a), the use of conflict constraints in the algorithm reduces them significantly (Figure 8.6b and 8.6c). Moving the points towards the centre of the quadnode is faster performance-wise, compared to checking for conflicts in the neighbouring quadnode, but introduces regularity to the generalised dataset (not shown in Figure 8.6).

This further conflict resolution comes with a performance cost as quadnode neighbours have to be traversed. Figure 8.7 shows average computation times for different cartographic conflict resolution strategies. For conflict resolution strategies including quadnode corners, scale changes take more time than if only cartographic conflicts in horizontal and vertical neighbours are being resolved, or all points are simply moved inside each quadnode.

This additional performance cost can be significantly reduced by storing inside each quadnode a reference to its neighbouring quadnodes, and storing the generalisation result inside the quadnode as shown in Section 6.5. That is, no further tree traversal is needed to select the neighbouring quadnode and only a simple collision check is required. This keeps the average computation time to around $300ms$ with a small increase in storage.

8.6. Data enhancement

Quadtree-based selection allows the retention of peculiarities of the point attributes across zoom levels. In Figure 8.8 a variant of value-based selection – selection based on local maxima of the attribute – illustrates how a particular attribute is retained through a series of scales.

In this case, selecting the points with maximum value for the red list status favours most endangered species (red dots), rather than maintaining the overall distribution of categories across the various zoom levels. Retaining the average of lichens per quadnode would effectively only show green spots, due to the skewed distribution of common lichens as opposed to endangered lichen species. Selection based on the local attribute value maintains, to a certain extent, the underlying spatial distribution of the point pat-

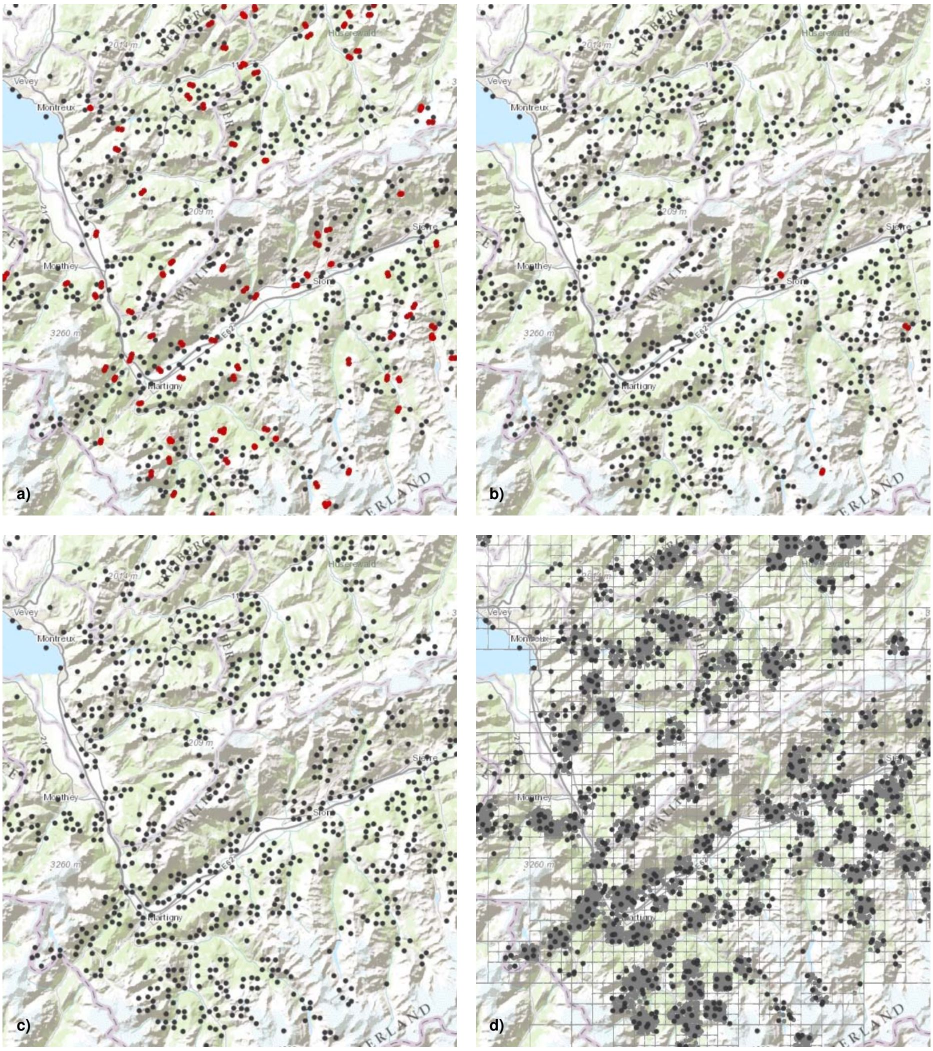


Figure 8.6.: Selection operator at zoom level 10: a) without conflict reduction; b) with horizontal and vertical neighbour conflict checks; c) including corner checks. d) Debuging view with the quadtree shown in the background and the source point distribution. Base map: ⓒ2013 ESRI ArcGIS Services World Topo Map

tern. Here it highlights clusters of most endangered lichens.

Each location with occurrence of lichens therefore shows the most endangered lichen per quadnode. That is, if at that location only less endangered lichens exist, the most endangered representative (green or yellow dot) of them is shown.

Local selection therefore effectively shows the local context among endangered lichens,

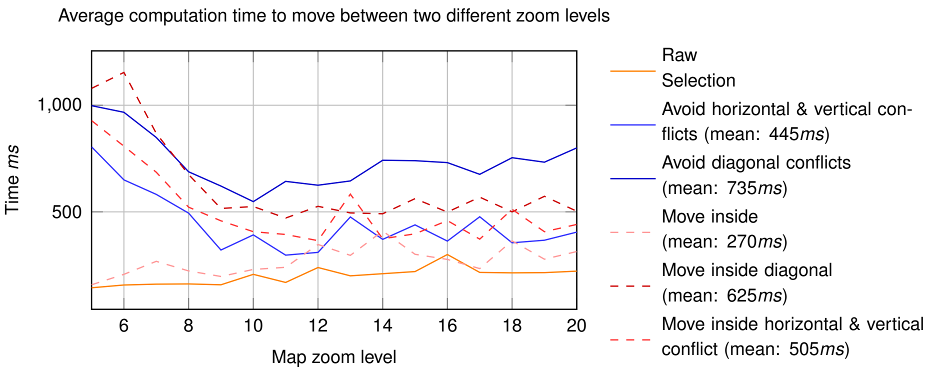


Figure 8.7.: Average computation time (for the entire SwissLichens dataset) to move between two different zoom levels for different options of quadtree-based selection if traversal of quadnode neighbours is required.

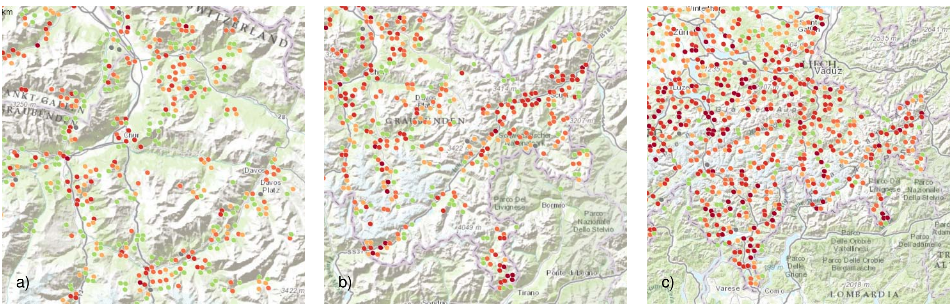


Figure 8.8.: Local value-based selection, retaining most endangered lichens, from zoom level 10 (a) to zoom level 8 (c). Base map: ©2013 ESRI ArcGIS Services World Topo Map

as opposed to only representing most endangered species (Figure 8.9b), retaining approximately the same number of points. In the case of global selection (Figure 8.9a) a base zoom level of 20 has been selected, as the depicted lichens have a rather small geographic footprint. The map represents approximately 1200 locations of the rarest lichens in Switzerland (though many are not visible due to overlaps). As they show a clustered distribution pattern, most of the points are not visible due to complete overlap. To retain the same number of points in the case of local selection, the parameterisation is such that it allows for partial overlap of the map symbols (Figure 8.9b). As can be observed by comparing the two maps it is apparent that global hotspots in the case of local selection (Figure 8.9b) are less pronounced. This effect can be reduced by applying clustering and density measures, retaining only those nodes that contain high amounts of points per quadnode.

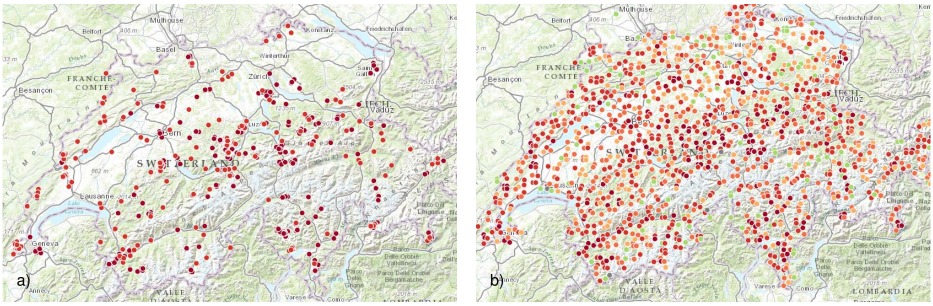


Figure 8.9.: a) Global *versus* b) local value-based selection, retaining most endangered lichens for zoom level 8. Colours range from least endangered in green to most endangered lichen species in red. Base map: ©2013 ESRI ArcGIS Services World Topo Map

Figure 8.10.: Aggregation by co-location of restaurants and parking in the city of Zurich with kernel density surface for a) map zoom level 13 and b) map zoom level 12. c) Kernel density difference between map zoom level 12 & 13, d) KDE difference of co-location at map zoom level 13 compared to the ungeneralised dataset. Base map: ©2013 ESRI ArcGIS Services Light Gray & CC BY SA 2013 OpenStreetMap.org

Another generalisation method especially suited for retaining peculiarities is *co-location filtering* introduced in Section 6.4.3. Based on combinatorial rules, co-location filtering retains map features located in the vicinity of each other. For example, restaurants with a nearby parking lot, a bus station or a cash point. Co-location depends on the map zoom level, as the smaller the map zoom level, the bigger the area and the higher the chance of finding co-located map features.

Figure 8.10 illustrates this difference for two consecutive zoom levels for the city of Zurich, retaining co-located restaurants and parking lots. The underlaid kernel density estimation (KDE) (Silverman, 1986) allows the visual comparison of the differences in the resulting density distributions between zoom level 13 (Figure 8.10a) and 12 (Figure 8.10b).

The difference between the two normalised density distributions (Figure 8.10c) highlights where the density pattern changed between the two zoom levels, due to the change of size of the aggregation to find co-located features. It shows places where further regions with co-located point features occur and where the density reduced. Further regions – marked in red – occur, for instance to the North (A) and to the South (B) and West (C) of the city centre, while a reduction of density can be observed in the city centre – marked in blue.

Compared to the ungeneralised baseline POI dataset, the density difference between the generalised and the source dataset shown in Figure 8.10d is evident at map zoom level 13, where the generalisation results of co-location are over-represented – in red – or under-represented – in blue – in comparison to the baseline. As the dataset used in this Figure only contains POIs which are either a place to eat or drink or a parking lot, it shows that in the old town of Zurich and in the central District 4 and 5 (D), parking places are relatively under-represented. While in Oerlikon (E), in the region around the main station and generally outside the core of the city centre there is more chance of finding co-located restaurants and parking possibilities.

8.7. Displacement measures

The displacement algorithm helps to further resolve spatial conflicts and thus allow retaining more point symbols than solely with point reduction operators. It tries to accommodate as many points as possible – keeping at most one point per quadnode – and displaces points, if the neighbouring quadnodes provide sufficient holding capacity for displacement (see Section 6.4.4).

Remaining overlaps can be removed by further solving boundary constraints as illustrated in Figure 8.6. The comparison of the results of mere point reduction (by centrality-based simplification) *versus* the displacement algorithm (Figure 8.11a, b) illustrates that displacement retains more points for the displayed zoom level. In Figure 8.11b the increase of points is in the range of 20 %. The number of points that can be additionally accommodated with the displacement operator highly depends on the spatial distribution of the dataset (cf. Figure 8.1).

On the other hand, Figure 8.11b shows that displacement has the effect of homogenising dense clusters, as dense areas get enlarged and thus affect the overall distribution

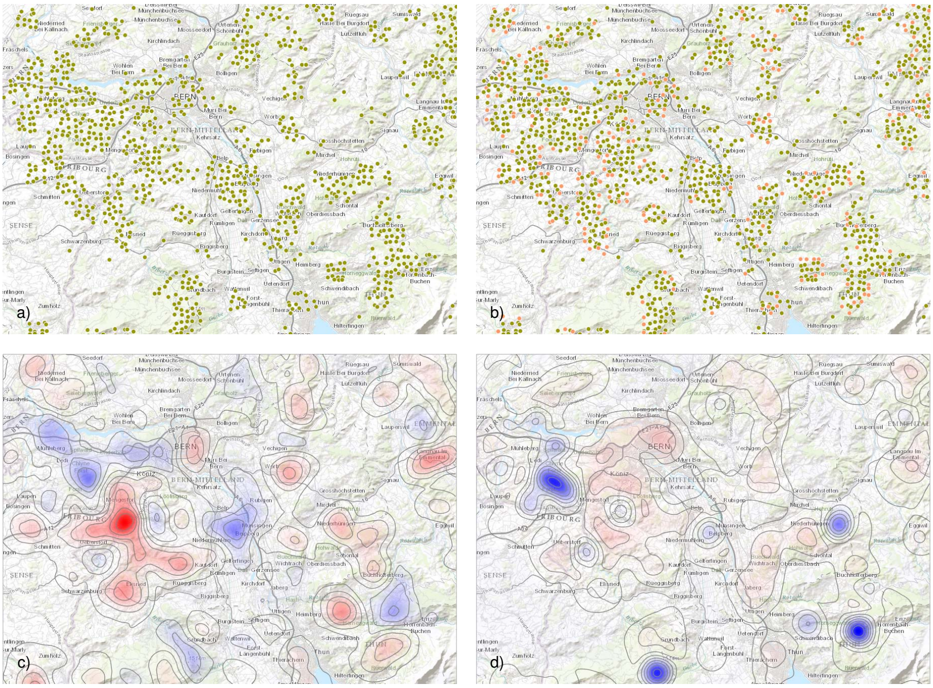


Figure 8.11.: a) Centrality-based simplification (1099 points), b) displacement applied after centrality-based simplification (1350 points), orange dots denote displaced point features, c) Kernel density difference between between a and b, d) Kernel density difference beween displacement and ungeneralised dataset. c,d) red denotes regions of increased densities and blue regions denote a decrease in density. Basemap: ©2013 ESRI ArcGIS Services World Topo Map

pattern. In Figure 8.11b the orange points denote those lichens that got displaced by the displacement operator. It shows that displacement is mainly effected on the boundary of larger clusters, where empty quadnodes are situated. A qualitative comparison between the spatial distribution of the two solutions (Figure 8.11a,b) shows that for displacement, the area covered by map symbols was enlarged, but the overall spatial distribution is maintained. Figure 8.11c highlights the difference of the kernel density estimation surfaces between both solutions. Red areas highlight where the density of the displacement generalisation is relatively higher than the case of centrality- based simplification. Blue areas, on the other hand, denote where the density decreased. Areas of increased density (red) are mostly situated on the border of clusters, whilst the cluster centres show a relative reduction in density (blue), due to the increase of density in the cluster border area, confirming the qualitative, visual observation.

A comparison of the displacement operator with the initial distribution of the dataset in Figure 8.11d shows a similar but less distinct pattern. Areas with high density clusters

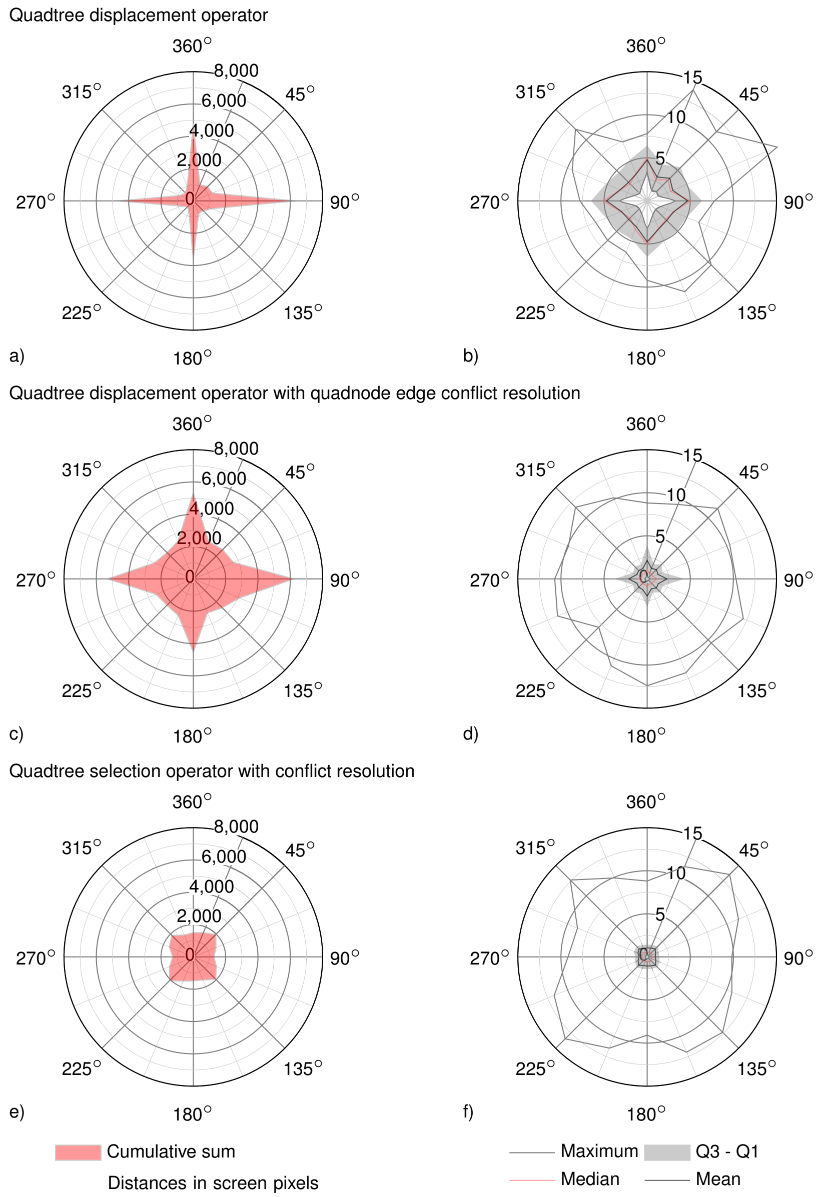


Figure 8.12.: Displacement vectors subdivided into 16 directional classes for the *displacement operator* a,b) without c,d) with conflict resolution and e,f) the *selection operator with conflict resolution* at map zoom level 11 of the SwissLichens dataset. a,c,e) Cumulative sum of the displacement vectors b,d,f) Distribution of the displacement vectors. Distances are shown in screen pixels

in the source dataset are significantly reduced (blue areas), while areas with a less dense coverage show a comparatively higher density. This is, on the one hand, due to regions with a high density of lichen observations in the source dataset that got reduced due to the selected proximity constraints in the quadtree-based generalisation. On the other hand, it shows that the quadtree-based generalisation, especially in the case of displacement, emphasises the overall local density pattern, rather than areas where globally the density is highest.

The characteristics of the displacement operator applied can be highlighted by plotting cumulative displacement vectors (in number of pixels) for each angle of displacement. Figure 8.12a shows nicely for the cumulative sum of the displacement vectors, that the algorithm – considering only horizontal and vertical neighbours (cf. Section 6.4.4) – hardly displaces points to diagonal neighbours. The majority of displacement angles are arranged in the four cardinal directions. Considering the distribution of the displacement vectors (Figure 8.12b), on average, points are displaced approximately half the symbol width of 10 pixels. Points displaced further than half the symbol width were located, instead, in the centre of the quadnode. Meanwhile the maximum length of the displacement vector at map zoom level 11 lies within the bounds of the quadnode width of 14 pixels, and the expected range of the diagonal of two quadnodes[1].

Using quadtree displacement with quadnode edge conflict resolution (Figure 8.12c) the overall sum of displacement increases. The remaining conflicts at the edges of the quadnodes are resolved by moving the points inside the quadnode (with a repel function). The distinct preference for horizontal and vertical displacement is reduced, due to the working principle of the edge conflict resolution, which resolves the remaining quadnode corner conflicts (Section 6.6). This is also reflected in the distribution of the displacement vectors, where the mean and the median of the displacement show a more even distribution for all directions (Figure 8.12d).

The difference in average displacement distances for horizontal and vertical directions against diagonal directions is slightly higher. The difference is higher, because the displacement distance of a point from one quadnode to the other is greater than the resolution of overlap between two points. This is not surprising, as in this setting the symbol size is smaller than the quadnode width, therefore the displacement distance from one quadnode to the other is higher than the displacement distance as a result of quadnode edge conflict resolution.

Figure 8.12e,f shows the cumulative sum and the distribution of the displacement vectors for quadtree-based *selection* with quadnode edge resolution by displacing points inside the quadnodes. Compared to the displacement vectors of the quadtree displacement operator, the amount of displacement is considerably smaller and the distribution shows a rather rectangular shape. The particular shape is a result of the edge conflict resolution, as conflicts are most likely to happen near the corner of the quadnodes.

Finally, the analysis of the displacement vectors does not show a skewed distribution towards one or two displacement angles, which indicates that the displacement algorithm

[1]The diagonal of two quadnodes of width n is $d = \sqrt{((2n)^2 + n)}$, for a quadnode of width 14 pixels the maximum possible displacement of 28.23 pixels should not be exceeded.

does not introduce a directional dependency.

8.8. Maintenance of spatial patterns

In this section the question of how well a particular generalisation algorithm maintains the overall point distribution is addressed. In other words, how does the distribution in the source dataset compare to the generalised solution. A good means of comparing the generalisation outcome of the different algorithms locally, is to investigate the local density variation. The overall areal density is a key property of any pattern of point located events (O'Sullivan and Unwin, 2010). An estimate of density is usually described as the number of points for a specified area. Density estimation, however, is sensitive to the definition of the area over which density is integrated. In the case of kernel density estimation, KDE, the estimate of density is described by means of a kernel function for each point event that attributes more weight to nearby events than distant ones (see O'Sullivan and Unwin, 2010). This definition of density estimation reduces the sensitivity to the reference area.

How well a generalisation algorithm maintains the underlying spatial point distribution pattern can be investigated by visually comparing the KDE of a point pattern for different zoom levels, or by quantitative analysis through calculating the difference between two kernel density estimations (see also applications of KDE for generalisation analysis in the previous section).

The following analysis describes absolute and relative variations of KDE for different map zoom levels after generalisation (Figure 8.14,8.15). Again, the SwissLichens dataset is used, this time with a focus on the Western part of Switzerland, to illustrate the influence of the quadtree-based generalisation algorithm (selection move) on the density distribution. This region is particularly interesting, as the spatial distribution of the point data in that region is highly uneven spatial distribution. A cartographically satisfactory solution by mere selection that maintains the global maxima in this case is hard to achieve. Another map representation and different generalisation operators may be more appropriate.

It is therefore seems fit to investigate the maintenance of spatial patterns of quadtree based generalisation. In the following, quadtree-based selection is used as a representative for the analysis. Due to the data-driven nature of the quadtree, the other quadtree-based operators show a comparable outcome for large changes in scale (cf. Section 8.7 on how the displacement operator affects the spatial pattern for small scale changes).

The overall distribution is shown in Figure 8.13b, where each cell of 5x5 pixels of approximately $500m^2$ with lichen occurrence is marked in green. The count of lichens for each cell is shown in shades of red, illustrating the density in that cell. From this figure it can be seen that the dataset itself is clustered, with one region containing very strong clusters to the West of Lausanne (A) and two further, but less pronounced, regions North-North-West of Lausanne (B) and in the West of Sarnen (C).

The number of observations in this region (> 55000 lichens) and the strong clustering in the described regions, however, makes the generalisation of the dataset necessary, but also leads in consequence to a compromise between maintaining the spatial coverage

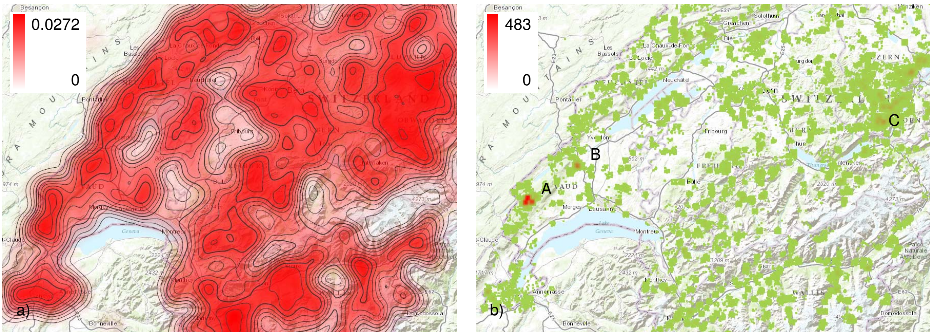


Figure 8.13.: a) Kernel density estimation for selection at zoom level 9, b) Pixel Counts of the source dataset, where each lichen observation covers an area of 5x5 pixels (colour range denotes the amount of observations from green to red), a Kernel density estimation with a crisp window (parzen window). Base map: ©2013 ESRI ArcGIS Services World Topo Map

and maintaining the global maxima (see also Figure 8.9a). The KDE for quadtree-based selection at map zoom level 9 (see Figure 8.13a) shows that the overall density distribution is maintained (in comparison to Figure 8.13b), whereas the identified high-density clusters dominating the distribution do not stand out.

The pure density difference between the generalised and the source dataset results in the three high-density clusters dominating the distribution (see Figure 8.13b). One option for removing the influence of these clusters is to compare the density distribution of the generalised map against the density of a one-pixel grid, storing for each cell – at most – one lichen occurrence. In other words, compare it to the highest possible resolution at that map zoom level and a symbol width of one pixel.

In this case of quadtree-based generalisation a similar analysis can be achieved by comparing the current zoom level to a higher map zoom level (content zooming) and a quadnode width close to one pixel. This density difference is illustrated in Figure 8.14d. Figure 8.14d displays the KDE difference between quadtree-based selection at map zoom levels 9 and 13. If at map zoom level 9 the quadtree generalisation up to zoom level 13 is shown (content zoom 13) the quadnode width for zoom level 13 is 0.8 pixels. The node width of content zoom level 13 at map zoom level 9 (see Chapter 4) can be derived from the following formula: $\frac{13}{16} = \frac{13}{13/2^{13-9}} = \frac{w_a}{2^{b-a}}$ the quadnode width w at map zoom level a divided by two to power of the difference of the zoom levels.

Figure 8.14 illustrates the evolution of absolute KDE differences between the generalised dataset at map zoom level 9 and zoom levels 10 – 13. The KDE density difference between two consecutive zoom levels is relatively low (Figure 8.14a), compared to the high differences that shows over four map zoom levels (Figure 8.14d). The sequence not only shows the absolute density difference between the respective map zoom levels, but also how the algorithm successively incorporates the density of the distribution according to the map zoom level. In other words, the algorithm reduces – as expected – the density where the highest densities occur and most conflict constraints have to be resolved. It

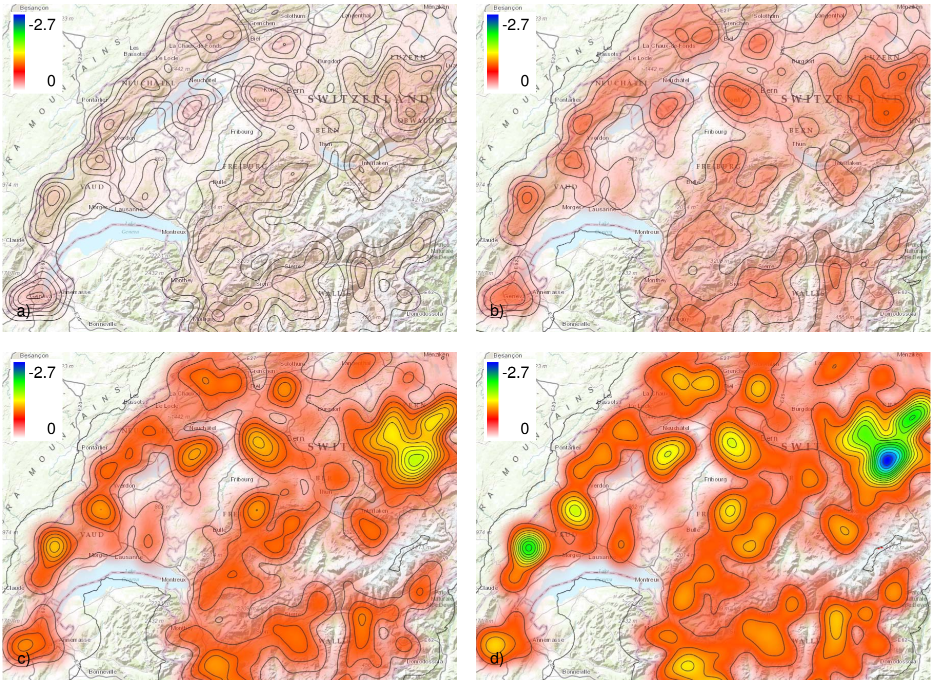


Figure 8.14.: Kernel density difference estimation for selection between zoom level 9 and zoom levels 10 – 13 (a-d) in absolute density differences. The colour ramp is the same for all four sub-figures. Base map: ©2013 ESRI ArcGIS Services World Topo Map

shows that the applied algorithm reduces the point density most at local peaks, where the highest densities are located and that overall, it maintains the underlying spatial pattern. It also shows that the algorithm hardly changes the point density in regions of low density, therefore the overall maxima of the point distribution in consequence, became less evident. This density equalising effect can be reduced by parameterising the generalisation operator differently (see Section 6.6).

An overview of the differences between normalised[2] density values with analogue zoom levels is shown in Figure 8.15 KDE. The colour ramp ranges from negative values in blue, where the density compared to the mean is reduced, to positive values in red denoting increased density values. Unlike in Figure 8.13 the range of colours does not represent the same range of values for all sub-figures, but is relative to the density values in each individual subfigure. Even though the four sub-figures are not directly comparable by the intensity of the colour values, one can observe that for small map zoom level differences (Figure 8.15a) the density differences are spatially dispersed, and show hotspots of under-representation in regions of high densities (cf. Figure 8.13a). Hotspot

[2]Normalisation based on the standard score or z-score $z = \frac{x-\mu}{\sigma}$, whereas μ denotes the mean of the density values and σ the standard deviation.

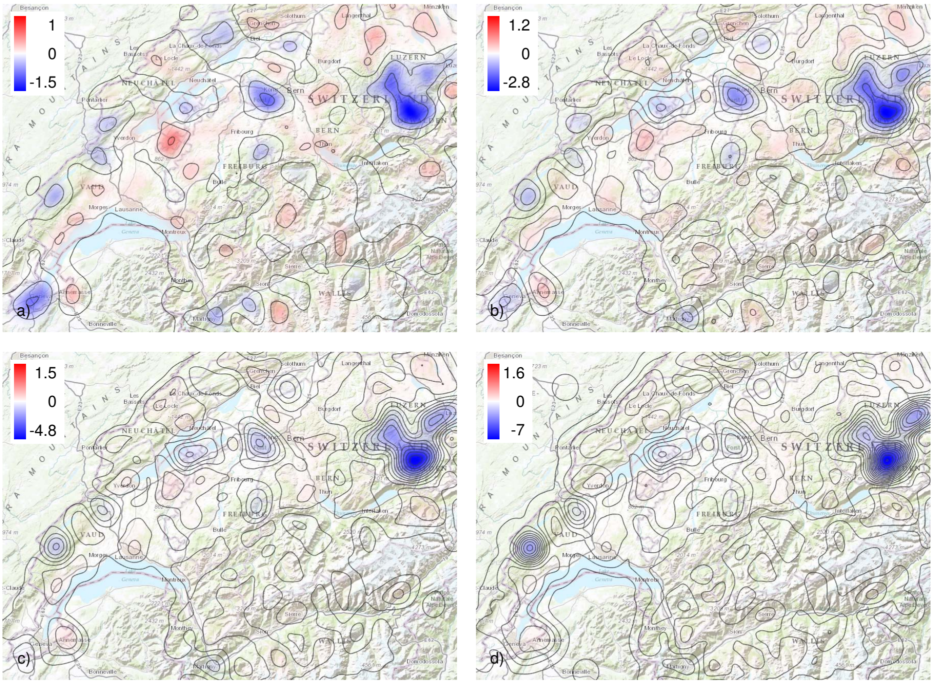


Figure 8.15.: Kernel density difference estimation for selection between zoom level 9 and zoom levels 10 – 13 (a-d) in relative z-score density differences. The colour ramp denotes relative negative density decrease in blue and positive increase in red for each sub-figure. To highlight the distribution of values, the colour ramp is tied to the range of density values in each sub-figure. Therefore the intensity of the colour values is not comparable between the sub-figures. Base map: ©2013 ESRI ArcGIS Services World Topo Map

over-representation tends occur along the borders of the high density regions.

This observation is in line with the observation made in analysing the effects of displacement (see Section 8.7). This effect is less distinct, the higher the difference in scale of the compared generalisation results (Figure 8.15d). At higher scale differences the under-representation in regions of high density dominates (blue), whereas, comparatively, the rest of the density differences is fading.

An alternative to comparing changes within the density distributions between the generalised and the source dataset, is to compute the CHI expectation surface (Wood et al., 2007). CHI statistics allow the comparison between observed and expected density distribution and the exploration of spatial variations in the observed distribution (in this analysis, the generalised map). CHI statistics (Equation 8.2) take the ratio between the density difference in each KDE grid cell and the square root of the expected density, in the same grid cell. The assumption is that the observed density is proportional to the expected density and if not, the observed values are either over- or under-represented compared to the expected distribution.

$$\chi = \frac{O(\lambda_{f,i}) - E(\lambda_{f,i})}{\sqrt{E(\lambda_{f,i})}} \tag{8.2}$$

χ	Chi expectation surface
$O(\lambda_{f,i})$	Observed density
$E(\lambda_{f,i})$	Expected density
$\lambda_{f,i}$	Density at map zoom level f at location i

While the observed density is equal to the density distribution of the generalised map, the expected density distribution remains to be defined. That is, the observed distribution should be proportional to the expected distribution of the dataset. The expected distribution can be defined in several ways, by either expecting the observed density to be proportional to the *source distribution* of the data, to the distribution *normalised by the Radical Law* (Töpfer and Pillewizer, 1966) or *normalised to the mean of the observed density*.

In the first case, by taking the source density distribution as the expected variable, for all values the CHI statistics show under-representation. This is due to the fact that the generalisation is a subset of the source dataset[3]. What is wanted in this analysis, however, is the expected density at the zoom level of the observed density values (map zoom level 9), to find regions of over- and under representation. In other words, regions where the generalisation algorithm reduced the density more or less than expected, respectively.

In the second case, the expected density is normalised to the expected number of points according to the Radical Law (see Equation 8.3). The normalisation of the values in the source distribution is such that the sum of all values equals the expected number of points according to the Radical Law. The difficulty of this approach, however, lies in the definition of the map scale of the source dataset M_a, respectively the base zoom level z_a.

$$\begin{aligned} E_{RL}(\lambda_{f,i}) &= \lambda_{a,i}\frac{n_{RLf}}{n_a} \\ &= \lambda_{a,i}\sqrt{\frac{M_a}{M_f}} \\ &= \lambda_{a,i}2^{\frac{z_f - z_a}{2}} \end{aligned} \tag{8.3}$$

$E_{RL}(\lambda_{f,i})$	Expected density based on the Radical law
$\lambda_{a,i}$	Density at source map a at location i
n_{RLf}	Number of derived objects based on the Radical Law
n_a	Number of objects at source map
M_a	Scale denominator of source map a
M_f	Scale denominator of target map f
z_a	zoom level $z = \log_2(a)$ (target zoom level)
z_f	zoom level $z = \log_2(f)$ (base zoom level)

In Figure 8.2 the base zoom level for the Radical Law is selected such that the area between the data reduction curves of the generalisation algorithm and the Radical Law is minimised. Here the base zoom level minimising the overlap region corresponds to zoom level 15.

[3]Normalisation to 1.0, as opposed to using the source dataset, shows over-representation for most of the density values, due to the highly uneven spatial distribution within the dataset.

Figure 8.16a shows the CHI expectation surface with the expected density normalised to the Radical Law with base zoom level 15. The CHI expectation map reveals large regions of under-representation. This is not surprising and in line with the data reduction curve (Figure 8.2) that suggests the retention of about four times more points for map zoom level 9 than are actually retained by the quadtree based generalisation algorithm.

A better way, therefore, would be the third case: to normalise the expected density surface to the mean of the observed density. Normalisation to the observed mean is shown in in Equation 8.4. Compared to the second Radical Scale, the ratio used corresponds to a base zoom level of 18.9[4].

$$\begin{aligned} \bar{E}(\lambda_{f,i}) &= \frac{\lambda_{a,i} \sum_{i=1}^{n} \lambda_{f,i}}{\sum_{i=1}^{n} \lambda_{a,i}} \\ &\cong \frac{\lambda_{a,i} n_f}{n_a} \end{aligned} \tag{8.4}$$

$\bar{E}(\lambda_f)$	Expected density based on same mean density
λ_a	Density at source map a at location i
λ_f	Density at target map f at location i
n_f	Number of objects at target map
n_a	Number of objects at source map

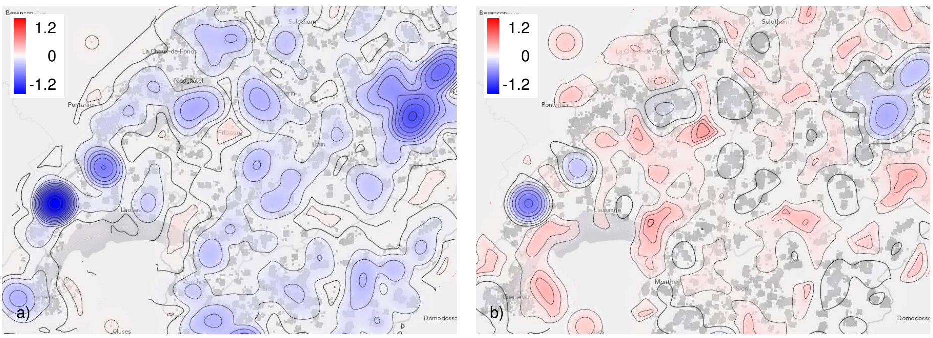


Figure 8.16.: Chi expectation surfaces for selection between observed and expected density distribution at map zoom level 9. Expected density distribution with a) Radical Law (Base Zoom Level 15) b) KDE normalised to the observed mean. Darker blue indicates less points than expected, while darker red indicates more points than expected. Value range: a) -1.28 – 0.23, b) -0.62 – 0.46. Base map: ©2013 ESRI ArcGIS Services

The resulting CHI expectation surface (Figure 8.16b) shows in relation to the mean, where the data has a higher/lower density than expected. Under-represented regions are clearly those regions with very high-density clusters. Over-represented regions, however, are situated where the density of the source distribution is low. The result shown is similar to the observations in Section 8.6, that compared to the underlying distribution, quadtree-based generalisation maintains the local spatial distribution. This generalisation approach emphasizes spatial coverage, maintains local maxima, but under-estimates global maxima. The under-estimation of the global maxima can be addressed by either setting the

[4] $z_a = -\log_2(ratio)2 + z_f$

parameterisation of the algorithm so as to better maintain *dense* clusters (see Section 8.10)) or by changing the map representation and use graduated symbols for high density clusters. Since any algorithm such a highly uneven spatial distribution is generalised, the trade-off between maintaining the spatial coverage of the data or the distinctness of the spatial clusters will remain a challenge and depends on the purpose of the map.

8.9. Homogenisation

The analysis of the homogeneity allows the analysis of how, and if, the spatial distribution of the map points exhibits an excessively homogeneous distribution after the generalisation. Several measures exist to analyse the spatial distribution, such as the nearest neighbour index (NNI) (Clark and Evans, 1954; O'Sullivan and Unwin, 2010). The nearest neighbour index R (Equation 8.5) is based on the average nearest-neighbour distance between points in a specified study area and determines if a point pattern is random, uniform or clustered. R values larger than 1, exhibits a uniform pattern (with $R = 2.15$ denoting a hexagonal pattern), while values below 1 represent clustered point pattern (with $R = 0$ indicating that all points are collapsed to a single point location).

$$R = \frac{\frac{1}{n}\sum_{i=1}^{n} d_i}{\frac{1}{2\sqrt{\frac{n}{A}}}} \tag{8.5}$$

R	Nearest neighbour index
A	Area
n	Number of points
d_i	Nearest neighbour distance

In the following, NNI analysis of the point collection of animal observations of the Swiss National Park is used to illustrate how the different quadtree-based generalisation algorithms change the spatial pattern during the generalisation process. The ungeneralised dataset has NNI of $R = 0.710$, which indicates a clustered spatial distribution. For taken NNI measures used here the reference area is the minimum bounding box of the base dataset. As the dataset is rather small and located within an area of approximatively 3.7 km^2 most of the generalisation operations occur between map zoom level 17 and 12. Note that the NNI is sensitive to the size of the area and edge effects.

For all plotted generalisation algorithms the distribution pattern changes from clustered to randomised, and in the extreme case of midpoint generalisation to a uniform distribution pattern. The degree to which a pattern changes to a uniform distribution depends on the generalisation algorithm used. In the case of the quadtree midpoint generalisation, for alls scales NNI is distinctively higher with an offset of approximately 0.14. This is due to the underlying regular pattern of the quadtree, as non-empty nodes are represented by a symbol in the centre of the node. NNI increases the smaller the map zoom level and the higher proportion of total quadnodes occupied.

In the case of quadtree-based selection, retaining those animal observations with the highest local observation counts – allowing for overlaps on quadnode borders – it shows that allowing for overlaps, the NNI is closer to the base distribution than if the selection operation resolves remaining conflicts. As NNI is based on the nearest-neighbour

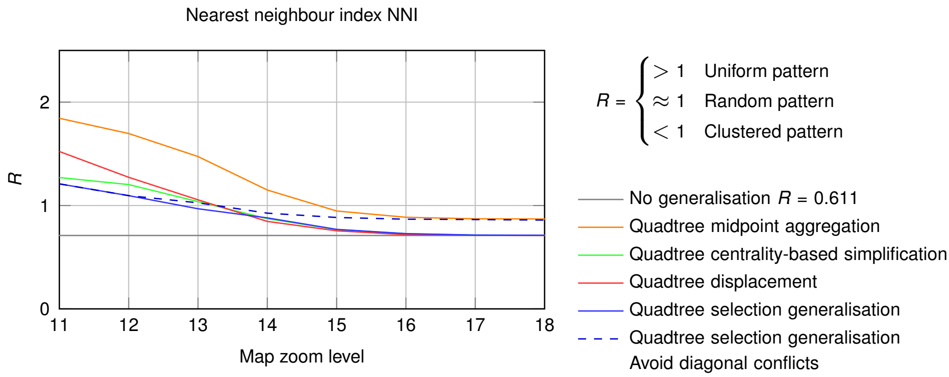


Figure 8.17.: Nearest neighour index (Clark and Evans, 1954) for animal observation data in the Swiss National Park for different map zoom levels and different generalisation algorithms

distances between points, it shows that a dataset without overlap – starting from a minimum distance between points, see Figure 8.19 – is always more regular or uniform than a pattern that allows for any distance between nearest points.

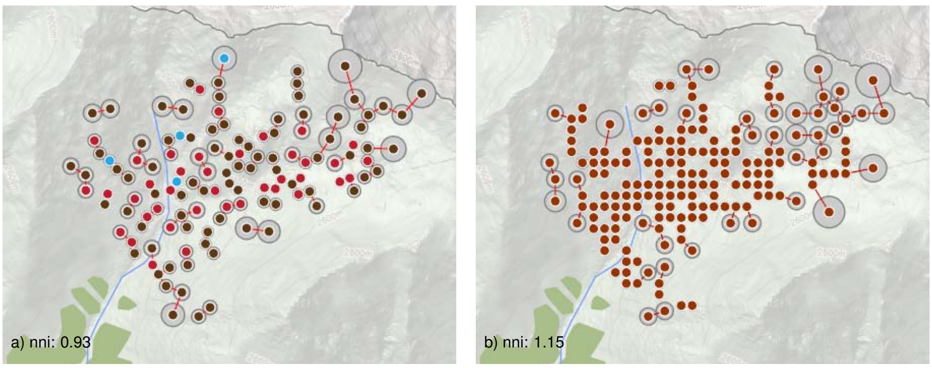


Figure 8.18.: Animal observation in the Swiss National Park, comparison of nearest neighbour distances for Quadtree selection a) and Quadtree midpoint aggregation for map zoom level 14. Red lines mark nearest neighbour connection, while the radius of the grey circle highlight half the distance to the nearest circle to give an impression of the nearest neighbour relations among the map features. Base map: ©2013 Google Maps & CC BY SA 2013 OpenStreetMap.org

The NNI is based on the ratio of the mean between expected and observed nearest neighbour distances in a specified area. As the index is based on the mean, characteristics of the nearest neighbour frequency distribution are not taken into account. Therefore the index conveys a more global explanation, as shown in the following example. A comparison between the generalisation outcome for map zoom level 14 of quadtree based selection with no overlaps between the map features and the rather regular quadtree midpoint-based aggregation, results in a NNI of 0.93 and 1.16 respectively shown in Figure 8.18. According to an index value of approximately 1, both spatial distributions exhibit a more or less random pattern. Visual comparison shows, however, that one is distinctively different from the other as Figure 8.18b exhibits a more regular and homogeneous pattern.

Unless explicitly required, a homogeneous pattern as depicted in Figure 8.18b is clearly not the desired generalisation outcome of a clustered spatial distribution. As the NNI reacts sensibly to outliers in a homogeneous pattern, the analysis of the histogram of the nearest neighbour distance distribution provides a further means of investigating the spatial distribution pattern. The respective histograms of the generalisation results are shown in Figure 8.19, and show a distinctly different distribution.

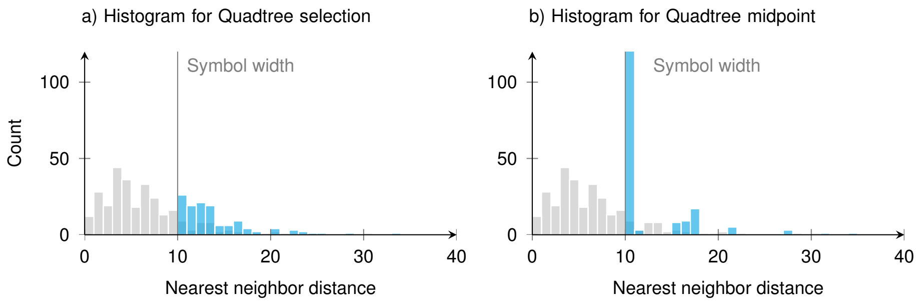
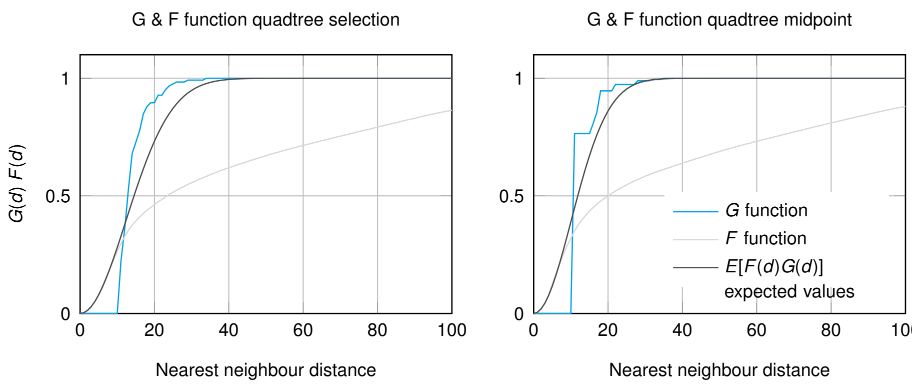


Figure 8.19.: Histogram of nearest neighbour distances between for a) Quadtree selection and b) Quadtree Midpoint aggregation of animal observation data in the Swiss National Park (see Figure 8.18). The histogram plotted in the background of both figures denotes the histogram of the nearest neighbour distances for the ungeneralised, source dataset.

For quadtree selection, the distribution of the histogram starts from a nearest neighbour distance of 10 pixels and decreases towards the higher end of the nearest neighbour distances, whilst for midpoint aggregation most of the observations are concentrated around the same value. Therefore analysing the distribution of the nearest neighbour distances and investigating the outliers provides a means of verifying if the distribution exhibits a certain regularity.

Distance-based point pattern analysis methods are an alternative to density-based methods (O'Sullivan and Unwin, 2010). They directly describe the distance relation between nearest neighbour events in a point pattern. G, F and K functions analyse the cumulative frequency distribution of the nearest neighbour distances and provide an insight into the frequency distribution of the nearest neighbours of a point pattern. The G function describes the fraction of nearest neighbour events within a pattern, while the F function describes the fraction of a random set of points inside the bounds of the point pattern to any event inside the point pattern (the more points the smoother the curve). For clustered patterns the G function rises quickly first, as it directly reflects the distance relation between points. As for the F function in the clustered case the function is likely to rise less sharp as the map space is comparatively empty (see O'Sullivan and Unwin, 2010). The



$$G(d) = \frac{no.[d_{min}(\mathbf{s}_i) < d]}{n} \qquad F(d) = \frac{no.[d_{min}(\mathbf{p}_i, S) < d]}{m} \qquad E[G(d), F(d)] = 1 - e^{-\lambda \pi d^2}$$

Figure 8.20.: G & F function of nearest neighbour distances for a) Quadtree selection and b) Quadtree Midpoint aggregation of animal observation data in the Swiss National Park (see Figure 8.18.

K function is similar to the G function but considers all distances for each point event to each neighbour in the point pattern, but is less easy to interpret.

Figure 8.20 presents the G and F functions and the expected distribution under the assumption of complete spatial randomness of the two generalised map results. In terms of investigating the regularity of the pattern, the G function is most insightful and shows – similarly to the histogram – distinct edges in the case of overly represented same distances, while the F function, which rather represents empty spaces of a point pattern, does not show clear differences between the two compared point patterns. The advantage of the G function in comparison to the histogram is that it does not rely on the binning of the data into frequency classes.

Thus the two compared maps both resolve conflict constraints, visible in the G function starting at the width of the symbol size. The difference between the two analysed generalisation methods in terms of homogenisation distinctly shows in the G function of the pattern, where the G function displays a smooth evolution over all nearest neighbour distances for quadtree-based selection. In the case of quadtree-based midpoint aggregation, however, the sharp edges in the G function indicate homogeneous, repetitive patterns.

8.10. Cluster maintenance

Previous sections showed that after generalisation the density in highly clustered areas is reduced and the overall coverage of the data maintained. This raises the question: are clusters maintained over the range of scales? And if the clusters vary in form and size, where do they change? Are they sufficiently distinct? If maintaining distinct clusters is required over large ranges of scales, which parameterisation supports this property?

This section investigates how the clusters change over a wide band of scales and how the parameterisations of the grouping constraints (see Section 6.6) can be set such that they are visually still perceivable as such clusters on small scales. That is, first a density-based clustering algorithm is needed to identify these clusters. The result of the clustering algorithm with the correct parameterisation should identify and mimic clusters on the map that map reader would visually detect. Very tiny clusters (sliver clusters) that would be represented as one point (i.e. a singularity) on the generalised map should be ignored in this analysis. With the help of the cluster algorithm – and a well adjusted parameterisation – the effects of the generalisation algorithm on the visual clusters are then studied and the feedback of a small survey group provides insights on the preferred settings of the grouping constraints.

Cluster analysis is a widely used technique for statistical and exploratory data analysis. In cluster analysis, clusters are defined as groups of similar objects, where the similarity among them is higher, compared to elements of other clusters. A visible cluster on a map can be defined as a group of objects, where the distance among the objects within the cluster is smaller than the distance to the rest of the objects of the map. Estivill-Castro (2002) argues that the notion of cluster depends on the cluster model and the applied properties, resulting in various different algorithms. Among these cluster models, models that are based on the density of the data points are probably best suited to analyse how well generalisation algorithms maintain clusters of points on the map.

Density-based clustering algorithms Density-based clustering algorithms, such as density-based spatial clustering of applications with noise (DBSCAN) (Ester et al., 1996) or its extension Optics (Ankerst et al., 1999), are good methods of extracting *dense* clusters of points on a map. DBSCAN identifies clusters with a density-based notion of clusters. Ester et al. (1996) state that the main reason why the algorithm recognizes clusters of arbitrary shape is that within the clusters the average density of points is considerably higher than outside. DBSCAN is based on the notion of *density reachability*, where a minimum number of points (*minPts*) that lie within an ε*-neighbourhood* of *core points* form a cluster (see Figure 8.21). The algorithm labels points not satisfying the density reachability criterion as *noise* points. The noise handling capability and the underlying density model allow the handling of clustered and sparse points, an indispensable property for the following analysis.

Unlike k-Means (MacQueen, 1967), DBSCAN does not require the specification of the number of clusters beforehand. The required parameters are *minPts* and ε*-neighbourhood*. Ester et al. (1996) propose the following heuristic to determine the parameters. For 2-dimensional data they suggest to set *minPts* to 4, and to estimate ε in a sorted *k-dist graph* (with $k = 4$). A sorted k-dist graph lists for each point the distance to the kth nearest neighbour in decreasing order. The first 'valley' or sharp bend in the graph denotes the distance of selection for the ε parameter, as the maximal k-dist value in the thinnest cluster of the dataset (Ester et al., 1996). That is, points with a 4-dist value lower than the ε threshold value are assigned to clusters, whereas points with a higher 4-dist value would be considered as noise. Ester et al. (1996) state that in general it is hard to detect the first valley automatically. However, depending on the dataset, finding the first valley is also

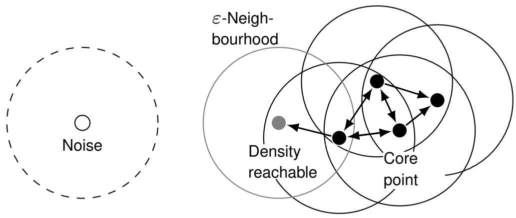


Figure 8.21.: Principle of the DBSCAN algorithm. Black points denote *core points*, where for each point at least $minPts = 3$ points lie within its ε*-neighbourhood*. *Density reachable points* (gray) lie within the ε-neighbourhood of other points, but do not fulfil the *minPts* criteria. Points that are not within the ε-neighbourhood of any other point are considered as *noise* (white).

visually not straightforward.

In determining *visual* clusters on a map, however, it seems appropriate to determine the ε*-neighbourhood* in relation to the symbol size of the point elements on the map. Figure 8.22 illustrates, with increasing ε-neighbourhood in relation to the symbol size, where the centre of a third point has to lie (gray area) in order to create a DBSCAN cluster (with minPts= 3) whilst violating proximity constraints.

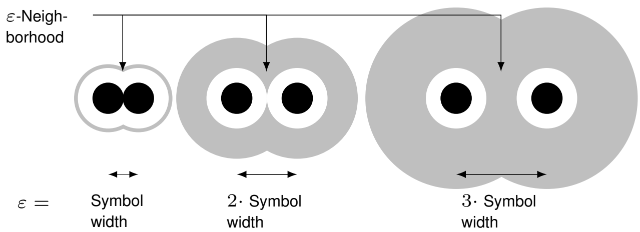


Figure 8.22.: Relation between symbol size and ε-neighbourhood definition. Gray areas denote regions where the centre of a third point needs to be located such that the cluster is detected if *minPts* $= 3$ and no proximity constraints are violated.

Parameterisation to define the 'visual' clusters To determine the ε*-neighbourhood* of the DBSCAN algorithm for the cluster analysis of the quadtree-based generalisation algorithms, a small set of participants ($n = 7$) was given the task of ranking a set of DBSCAN cluster maps with varying ε*-neighbourhood*. The participants were selected from the PhD students of the GIScience Center at the Department of Geography who had received cartographic training. The participants were given a map with the source ungeneralised dataset and four maps with DBSCAN clusters and varying ε-distances for each of the cluster maps. The ε parameter was set in relation to the symbol size (8 pixels) of the depicted points, which was 1 (a), 1.5 (b), 2 (c), 2.5 (d) and 3 (e) times the symbol size (i.e. 8, 12, 16, 20 and 24 pixels). Each user was asked to rank the results, based on how well the DBSCAN clusters represent and summarise the clusters they observes on the ungeneralised map.

The map used is shown in Appendix C, along with the associated questionnaire and answers. It depicts eating places (such as restaurants, cafés and bars) in the canton of Zurich at approximately map zoom level 11. The region is well known to the participants and well suited for the cluster analysis, as it contains differently sized clusters (smaller and larger cities) and also contains a large set of points that do not belong to any cluster (e.g. a restaurant on the hill top).

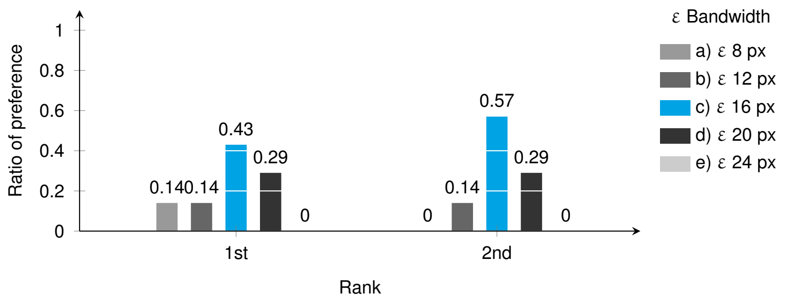


Figure 8.23.: Ranks attributed by a small group of participants ($n = 7$) on the preferred ε-neighbourhood. The four ε-distances where 8,12,18,20 and 24 pixels, whereas the symbol size in all four maps was set to 8 pixels.

The result of the study (Figure 8.23) shows that for this map, the preferred ε-*neighbourhood* distance lies around two times the symbol size. The smaller ε-distance was rejected by the participants as being too detailed, whereas the largest ε-distance in the group was considered to create too large a cluster by adding many sub-clusters to one large group (e.g. in region around the city of Zurich). The participants also had the possibility of adding some thoughts and comments on why they had chosen a specific ranking.

Among the various oral and written comments, one user noted that Zurich as one big cluster does not really make sense, and that he would have preferred a further hierarchical clustering within the city. He points out one drawback of the DBSCAN algorithm, which is that it cannot cluster datasets that have a large variation in densities. In this analysis, however, DBSCAN is well suited, because the algorithm clusters along the border between what is considered dense and what is sparse, in a binary fashion. This distinction between dense clusters and sparse regions, with a user-defined ε-*neighbourhood* distance, then allows the comparison of the DBSCAN clusters of the generalised dataset against the source dataset. And it therefore allows the analysis of how well clusters are maintained by the generalisation algorithm, or which parameterisation is optimal for the maintenance of clusters especially at small map zoom levels, where the algorithm tends to homogenise the she spatial arrangement of points (see Section 8.9).

A second comment was directed at the given case study (restaurants in the canton of Zurich), stating that the selected ranking is tied to the given topic and that another topic – such as hotels – could have resulted in another ranking. In order to apply the preferred ε-*neighbourhood* distance found in this small study more generally and for further analysis, this statement remains to be tested. Here, in the following cluster analysis, the same dataset, scale and region are used for comparability of the results.

Cluster analysis: Definition of terms In the following some terms are introduced to clarify the discussion of the analysis:

Conflict constraints denote how the quadtree-based generalisation algorithm treats conflict constraints on quadnode border. That is: *none*, if no border conflict reduction is applied, *avoid* and *move*, if in case of border conflicts one point is removed or moved, respectively. *Fill* as a third option (cf. Section 6.6).

Grouping constraint d, p are two parameters and local quadnode measures (see Table 6.3 on page 75) that allow the emphasis of clusters in the generalised map. The grouping constraints are especially suited for small map zoom levels. The *depth grouping constraint*, d, is based on the maximum depth reached by the leaf nodes of a quadnode. The *point-based grouping constraint*, p, sets the requirement that the points contained in a quadnode are only retained for generalisation if the number of points is at least the number of points defined in the grouping constraint p.

Symbol-to-node ratio denotes a parameter that allows the setting of the ratio between the symbol size and the quadnode width. The symbol-to-node ratio affects the overall density of the generalised map. A symbol-to-node ratio of 1:1 sets the quadnode width equal to the symbol size, a symbol-to-node ratio of 2:3 sets the quadnode width to $1.5\times$ the symbol size (Figure 8.24).

Clusters indicates the number of clusters found by the DBSCAN algorithm (minpts: 3, ε: 16 pixels, $2\times$ symbol size)

Noise denotes the number of points classified as noise by the DBSCAN algorithm.

Total points denotes the number of points retained by the generalisation algorithm.

Cluster ratio is the ratio between the number of clusters in the generalised dataset to the number of clusters in the source data set at map zoom level 11.

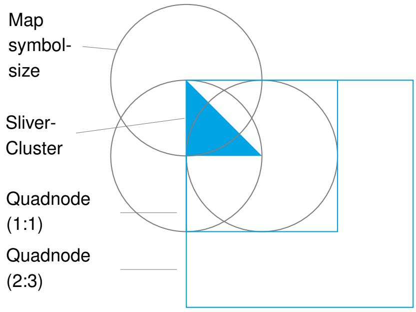
Cluster sliverless ratio is the ratio between the clusters in the generalised dataset to the clusters in the source dataset, without clusters with convex hull smaller than half the symbol area.

Noise ratio denotes the ratio between the number of noise points and the total number of points (Figure 8.24).

Points ratio denotes the ratio between the number of points of the restricted generalisation algorithm to the number of points of the unrestricted generalisation algorithm.

The cluster analysis is structured as follows. First, a comparative cluster analysis is performed at the same map zoom level as used in the questionnaire. The analysis investigates a selected quadtree-based generalisation algorithm with two different symbol-node ratios applied and compares it to the cluster analysis of the source dataset at map zoom

Figure 8.24.: Schematic illustration of the used symbol-to-node ratio 1:1 and 2:3 and the concept of sliver clusters



level 11. The second part of the analysis then concentrates on quadtree-based selection with different clustering over two consecutive map zoom levels (map zoom level 10 and 9), and sets the cluster analysis in relation to the cluster results in map zoom level 11.

Cluster analysis: comparison of algorithms at map zoom level 11 Of the object-directed generalisation algorithms (see Chapter 6) a quantitative point reduction (selection) and qualitative point displacement (displacement) are selected to perform the cluster analysis at map zoom level 11. For each algorithm different conflict constraints (none, avoid and move) and two symbol-to-node ratios ($1:1$, $2:3$) are applied.

Table 8.3 shows the result of the DBSCAN algorithm for the source dataset and the different generalisation settings. Each row lists the result of one DBSCAN cluster analysis, of which the *cluster ratio*, *cluster sliverless ratio* and the *noise ratio* are listed in a bar chart for comparison. The first entry lists the result for the source dataset, with 81 different clusters found and a noise ratio of 0.11.

A considerable number of these clusters have a quite small footprint with few points forming a cluster. These are restaurants in close vicinity, which at the analysed map zoom level would not appear as a cluster. A point reduction generalisation algorithm would most likely represent these clusters with a single symbol. Quadtree-based generalisation algorithms (except displacement) represent these small clusters as one point, if they fall into one quadnode (see Figure 8.24). These small *sliver clusters*, however, distort the expected number of clusters to be maintained by a generalisation algorithm at the investigated map scale. Hence, for the cluster analysis these sliver clusters are not considered. In this analysis a cluster is considered a sliver cluster if the area of the convex hull formed by the points of the cluster is smaller than an eighth of the area covered by the squared symbol size (Figure 8.24). Therefore, for the source dataset without the sliver clusters only 51 clusters remain resulting in a noise ratio of 0.15 of 2065 points in total. The number of clusters without the sliver clusters denotes the expected number of clusters at a specified map zoom level.

Purely analysing the number of clusters has to be considered with care. Even though the number of clusters reduces with increasing map scale, in certain cases clusters split

Table 8.3.: Cluster analysis for map zoom level 11 with DBSCAN (minpts: 3, ε: 16 pixels, $2\times$ symbol size).

Map zoom 11		Clusters	Cluster ratio	Cluster sliverless ratio	Noise	Total points	Noise ratio
1	none	-	1.00	-	221	-	0.11
	" sliverless	57	0.70	-	312	2065	0.15
Generalisation algorithms							
Quadtree symbol node ratio 1:1							
Selection							
2	None	56	0.69	0.98	261	755	0.35
3	Move*	54	0.67	0.95	252	755	0.33
4	Avoid[1]	40	0.49	0.70	293	543	0.54
5	Fill	39	0.48	0.68	284	566	0.50
Displacement							
6	None	66	0.81	1.16	204	901	0.23
7	Move*	64	0.79	1.12	213	900	0.24
Quadtree symbol node ratio 2:3							
Selection							
8	None	39	0.48	0.68	303	592	0.51
9	Move	39	0.48	0.68	293	595	0.49
10	Avoid	33	0.41	0.58	318	498	0.64
11	Fill	39	0.48	0.68	287	591	0.49
Displacement							
12	None	58	0.72	1.02	259	735	0.35
13	Move[2]	56	0.69	0.98	251	735	0.34

* Preferred generalisation solution
[1] Figure 8.25a, [2] Figure 8.25b

up into a set of smaller clusters (see Figure 8.25a and the new-formed clusters in the city of Winterthur in the NE corner). Therefore a change in the number of observed clusters is a combination of appearing[5], disappearing, merging and splitting clusters.

Table 8.3 shows the differences between a wider symbol-to-node ratio of $2:3$ and a tight $1:1$ parameterisation. A wider symbol-to-node ratio (Table 8.3, line 8-13) generally reduces the cluster ratio and increases the noise ratio, compared to the $1:1$ symbol-to-node ratio, except in the case of selection (fill). Selection (fill), however, is a special case, as the fill option allows the possibility of filling up the quadnode with more than one point, which explains why the cluster numbers hardly change.

Comparing the generalisation operator selection to displacement shows in line with the observation in Section 8.4, that displacement retains more clusters than selection. Similarly, within the different conflict constraint options, selection without any conflict

[5] In the case of generalisation, however, clusters should not appear (i.e. new clusters not be formed) in the course of a generalisation process.

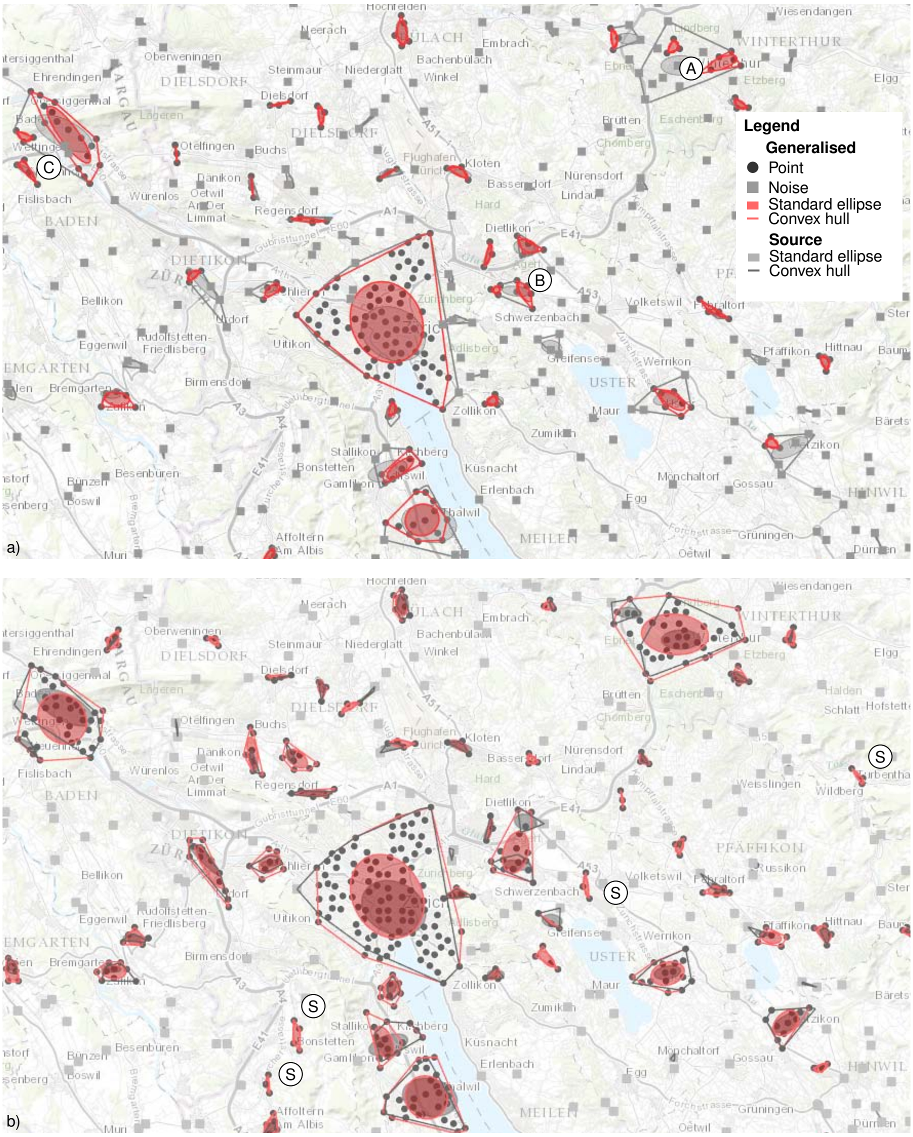


Figure 8.25.: Cluster analysis for map zoom level 11, for generalisation operation a) selection (avoid) and a symbol-to-node ratio of 1:1, b) displacement (move) and a symbol-to-node ratio of 2:3.

constraints or the selection (move), retains more clusters and a noise ratio that better reflects the source dataset. However, in these cases conflict constraints may not be fully resolved, as was shown in Section 8.4.

Figure 8.25a, b shows two sample cluster analysis maps corresponding to lines 4 and

13, respectively in Table 8.3. Both figures differ in the symbol-to-node ratio and the generalisation operator applied, and are therefore not directly comparable. Red colour marks the convex hull of the cluster, while the red ellipse denotes its standard ellipse.

Generalised points within clusters are marked as dark grey points and noise is marked with grey squares. The clusters of the source dataset are marked with a grey convex hull and standard ellipse. This overlay permits assessing how the clusters changed in the course of the generalisation process, and if they maintain size, disappear, merge or split.

In the case of *selection* (Figure 8.25a) clusters tend to reduce in size and split. Clusters of restaurants split in the cities Winterthur (A), Dübendorf (B) and Baden (C). The shape of the clusters, if they are not split and not considerably smaller, is more or less maintained.

The analysis for the generalisation operator *displacement* (Figure 8.25b) shows that the shape of the clusters is generally better maintained, even though the symbol-to-node ratio in Figure 8.25b is lower and therefore implies a reduced density. Due to the working principle of the displacement operator, clusters merge rather than split. Therefore only a few clusters are completely removed, while in addition a set of four sliver clusters (S) is being maintained.

Cluster analysis: comparison across three map zoom levels Table 8.4 and 8.5 show the results of the DBSCAN analysis for *selection* for map zoom levels 10 and 9. The two tables compare the generalisation operator selection with and without conflict constraints and the effect of different grouping constraints d, p on the cluster analysis.

For map zoom level 10 selection (none) without grouping constraints (Table 8.4 line 1) results in a cluster sliverless ratio 0.71 and a noise ratio of 0.15. Compared to map zoom level 11 there are less clusters detected, whereas the noise ratio stays the same as the raw point pattern. That is, there are less clusters, but they cover a larger geographic area.

Selection (move, Table 8.4, line 8) and Selection (avoid, line 15) both show a similarly reduced cluster sliverless ratio in constrast to selection (none). The noise ratio in selection (avoid) raises to 0.3, compared to 0.13 in selection (move).

A closer look at the effect of the grouping constraints p, d shows with increasing parameter values (1-3) a reduction in the cluster sliverless ratio and an increase of the noise ratio, while the points ratio is reduced. A notable difference between the two parameters is the rate at which the ratio changes, reflecting their different working principles. In the case of *point-based grouping constraints* p (see Section 6.6) the ratio changes in a seemingly continuous way. While with *depth grouping constraint* d (see Section 6.6) the ratio instead changes in a discontinuous fashion.

The reason for the rather discontinuous change is given by the fact that the *depth grouping constraint* depends on a minimum depth that the leaf nodes of a quadnode have to reach. That is, in a point region quadtree, two close points may generate a quadtree with a high depth. These points are therefore retained by the *depth grouping constraint*, causing a seemingly discontinuous change in the density-based clustering, as the points are not removed based on the density criterion. Conversely, this is the case with the *point-based grouping constraint*.

Table 8.4.: Cluster analysis for map zoom level 10 with DBSCAN (minpts: 3, ε: 16 pixels, $2\times$ symbol size).

Selection	Map zoom 10 width quadtree symbol node ratio 1:1						
	Conflict constraint	Grouping constraint *d*	*p*	Cluster ratio	Cluster sliverless ratio	Noise ratio	Points ratio
1	None	-	-	0.37	0.71	0.15	1.00
2		0	1	0.31	0.60	0.16	0.77
3		0	2	0.22	0.43	0.32	0.45
4		0	3	0.12	0.24	0.40	0.28
5		1	1	0.22	0.43	0.32	0.45
6		2	1	0.17	0.33	0.39	0.36
7		3	1	0.16	0.31	0.39	0.28
8	Move	-	-	0.30	0.57	0.13	1.00
9		0	1	0.26	0.50	0.16	0.77
10		0	2	0.20	0.38	0.30	0.45
11		0	3	0.14	0.26	0.37	0.28
12		1	1	0.20	0.38	0.30	0.45
13		2	1	0.17	0.33	0.35	0.36
14		3	1	0.17	0.33	0.36	0.28
15	Avoid[1]	-	-	0.28	0.55	0.30	1.00
16	$*$[2]	**0**	**1**	0.26	0.50	0.32	0.73
17	[3]	0	2	0.11	0.21	0.56	0.47
18	[4]	0	3	0.05	0.10	0.64	0.28
19	$*$	**1**	**1**	0.11	0.21	0.56	0.47
20		2	1	0.06	0.12	0.64	0.38
21		3	1	0.05	0.10	0.69	0.29

* Preferred generalisation solution
[1] Figure 8.26a, [2] Figure 8.26b, [3] Figure 8.26c, [4] Figure 8.26d

Figure 8.26a-d shows an example of the progression of the *point-based grouping constraint* p (Table 8.4, lines 15 – 19), with the same map symbolisation as in Figure 8.25. Figure 8.26a denotes selection (avoid) without grouping constraint, while subfigures c-d show point-based grouping constraint with $p = 1, 2, 3$. Starting with subfigure a, compared to source clusters (in grey) DBSCAN clusters of the generalised map cover a larger area. In the subsequent subfigures b-d the area of these clusters is reduced, or the clusters split or disappear. The largest three clusters of restaurants remain with all three settings. Small, and then mid-sized, clusters are subsequently removed with increasing parameter values.

The same analysis of increasing the grouping parameter at map zoom level 9 (Table 8.5) shows a similar behaviour for the noise- and the points ratio. That is, for increasing parameter values the noise ratio increases and the points ratio decreases. The number of clusters, however, is generally low, and clusters cover a large area on the map, if not the whole map.

Precisely, at this zoom level the NNI reaches the value of a random pattern, unlike the maintained clustered pattern at map zoom levels 10 and above. The grouping constraints place emphasis on the largest clusters and hence reduce sparsely spread points in areas where there are no clusters. Thus, by increasing the values of the grouping constraint, the number of clusters is at first low, then reaches a peak, and finally decreases again.

For selection options *none* and *move*, the highest cluster number is reached for $p = 4$ (Table 8.5, lines 5, 18) and $d = 2$ (Table 8.5, lines 9, 22) . For selection (avoid, 8.5, lines

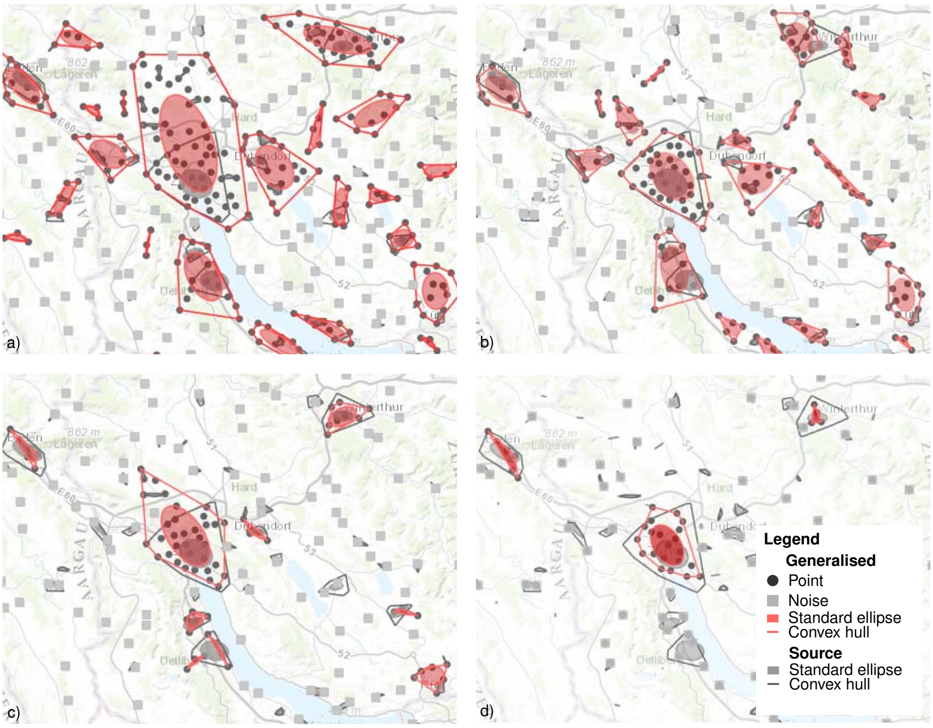


Figure 8.26.: Cluster analysis for map zoom level 10, for generalisation operation selection (avoid); a) without; and with point based grouping constraint b) $p = 1$; c) $p = 2$; d) $p = 3$ and a symbol-to-node ratio of 1:1

.

29, 35) the highest cluster number is in both cases the grouping constraint with value 2. The noise ratio is, in the cases with the highest number of clusters (and thus highest cluster sliverless ratio), similar or even lower than the noise ratio found at map zoom level 11. A similar or lower noise ratio than expected (here the noise ratio at map zoom level 11) results in visually less distinguishable clusters. Thus, highlighting the most important clusters on a small map scale by increasing the grouping constraints, results in a higher noise ratio than in the reference map. The emphasis on large clusters also highly depends on the map purpose.

Figure 8.27 shows the distinct change of cluster sizes and extents at map zoom level 9, triggered by different values for the grouping constraints p, d. Subfigures a,d) present selection (avoid) and (move) without any grouping constraints, whereas subfigures b,c and e,f show the corresponding cluster result with grouping constraint $p, d = 4$. The different working principle of the conflict and grouping constraints affect how the clusters are maintained. All four subfigures with grouping constraints (b,c,e,f) maintain the most dominant clusters. Of these subfigures, 8.27c maintains more clusters and their shapes fit

Table 8.5.: Cluster analysis for map zoom level 9 with DBSCAN (minpts: 3, ε: 16 pixels, $2\times$ symbol size).

Selection Map zoom 9 width quadtree symbol node ratio 1:1

	Conflict constraint	Grouping constraint *d*	*p*	Cluster ratio	Cluster sliverless ratio	Noise ratio	Points ratio
1	None	-	-	0.02	0.09	0.01	1.00
2		0	1	0.02	0.09	0.00	0.93
3		0	2	0.06	0.22	0.04	0.62
4		0	3	0.09	0.30	0.14	0.42
5		0	4	0.12	0.43	0.22	0.30
6		0	5	0.06	0.22	0.35	0.22
7		0	6	0.04	0.13	0.46	0.16
8		1	1	0.06	0.22	0.04	0.62
9		2	1	0.12	0.43	0.09	0.50
10		3	1	0.11	0.39	0.18	0.42
11		4	1	0.11	0.39	0.20	0.33
12		5	1	0.07	0.26	0.35	0.23
13		6	1	0.05	0.17	0.43	0.15
14	Move	-	-	0.01	0.04	0.00	1.00
15		0	1	0.01	0.04	0.00	0.95
16		0	2	0.05	0.17	0.04	0.62
17		0	3	0.06	0.22	0.13	0.42
18		0	4	0.09	0.30	0.23	0.30
19		0	5	0.06	0.22	0.30	0.22
20		0	6	0.05	0.17	0.40	0.16
21		1	1	0.05	0.17	0.04	0.62
22		2	1	0.09	0.30	0.08	0.50
23		3	1	0.09	0.30	0.17	0.42
24		4	1	0.07	0.26	0.19	0.33
25		5	1	0.07	0.26	0.28	0.23
26		6	1	0.05	0.17	0.43	0.15
27	Avoid	-	-	0.02	0.09	0.03	1.00
28		0	1	0.04	0.13	0.01	0.95
29		0	2	0.12	0.43	0.11	0.69
30		0	3	0.09	0.30	0.29	0.49
31	*	**0**	**4**	0.06	0.22	0.47	0.38
32		0	5	0.04	0.13	0.53	0.26
33	*	**0**	**6**	0.02	0.09	0.68	0.18
34		1	1	0.12	0.43	0.11	0.69
35		2	1	0.14	0.48	0.25	0.58
36		3	1	0.10	0.35	0.41	0.49
37	*	**4**	**1**	0.10	0.35	0.40	0.39
38		5	1	0.05	0.17	0.52	0.27
39		6	1	0.01	0.04	0.71	0.18

* Preferred generalisation solution

more tightly to the expected clusters (grey), whereas for subfigures b,e,f the cluster area expands, especially in the case of selection (move) in subfigures e and f.

Which constraint is eventually selected depends in practice on a qualitative judgement as well as on the map usage envisioned and the preferences of the user. At zoom level 9, a change of the map representation for small map scales may also be considered. Thus, to gain further insight into what settings of grouping constraints are preferred, a second survey with the same group of seven participants was conducted. The corresponding questionnaire is listed in Appendix C.2.

In the questionnaire the participants were asked to rank generalisation results according to their preferences, in a similar way as in in the first survey. The presented maps correspond to the analysed maps for selection (avoid) at map zoom levels 11, 10 and 9, with varying grouping constraints.

In the first task for map zoom level 11, three maps were compared with the following grouping constraints: no grouping constraints, $p = 1$, and $d = 1$. Of these the solution preferred by the participants was $p = 1$.

In the second task for map zoom level 10, for each grouping constraint p and d maps with parameters ranging from 1 to 4 were compared. For subtask (a) grouping constraint $p = 2$ was the solution ranked first by the participants, while for subtask (b) a grouping constraint $d = 1$ was ranked first.

In the last task at map zoom level 9, six maps were compared for each grouping constraint. The parameter values in the two subtasks for grouping constraint p and d are $1, 2, 3, 4, 6, 8$. In the case of grouping constraint p for subtask (a), $p = 4$ and $p = 6$ were ranked equally as best solutions, whereas in subtask (b) with grouping constraint $d = 4$ ranked best.

The complete ranking tables for the tasks are presented in the Appendix C.2. The preferred solutions for map zoom levels 10 and 9 are marked with an asterisk in Tables 8.4 and 8.5. What can be observed is that with more options the participants have to select and smaller map scale the ranking results are less distinct (compare between map zoom level 11 and 9). However, it is interesting to note that the participants show an agreement on the worst map rather than on the best generalisation solution.

Regarding the parameterisation it can be said that the smaller the map zoom level, the higher the noise ratio has to be, in order to keep the clusters distinguishable and provide an acceptable solution to the map user. It is not the highest *cluster sliverless ratio* that defines the best grouping constraint, but a combination between *noise ratio* and *cluster ratio*. It can be also said that less importance has to be given to how well regions with sparse point data are maintained with respect to clustering, as these regions basically remain untouched over a range of zoom levels. However, in order to keep the clusters in the dataset at small map scales distinguishable, the noise ratio has to be increased whilst the points ratio and the cluster sliverless ratio is decreased.

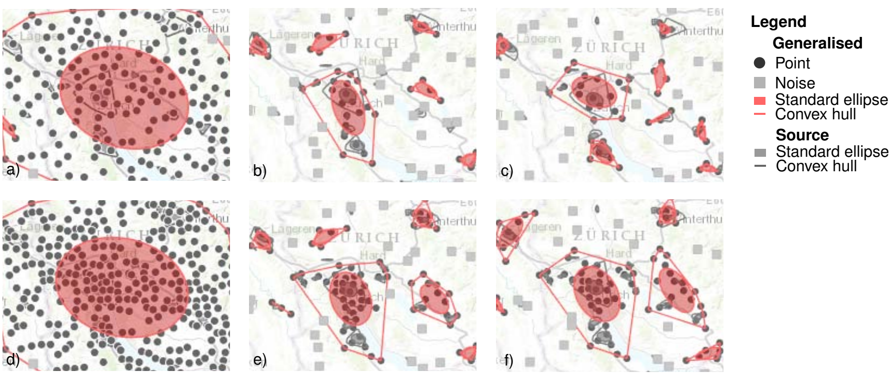


Figure 8.27.: Cluster analysis for map zoom level 9, for generalisation operation selection (avoid); a) without grouping constraints; b) $p = 4$; c) $d = 4$; and selection (move); d) without grouping constraints; e) $p = 4$; f) $d = 4$; and for all a symbol-to-node ratio of 1:1.

8.11. Concluding remarks

The cartographic analysis covers selected aspects of the generalisation results, as well some strengths and weaknesses. The analysis of data reduction applied to different datasets show that the data reduction curve does not follow the Radical Law (Töpfer and Pillewizer, 1966), but rather follows an S-shape, due to the underlying spatial distribution of the data, that is not directly reflected in the formula of the Radical Law. The data reduction curves further indicate, in their steepest part, at which zoom level most cartographic conflicts occur, which is confirmed by analysing the occurrence and reduction of cartographic conflicts per zoom level.

Comparing local and global selection criteria indicates that through local selection and data enhancement the context of the spatial distribution is maintained. A comparison between simplification and displacement operators demonstrated the additional number of points that can be accommodated by the displacement operator. And finally, the analysis of the spatial point distribution showed that the spatial arrangement is mainly changed at locations with high point density. Low density regions may be over-represented, which can be tackled by a different parameterisation of the algorithm by the grouping constraints.

Chapter 9.

Workflow and application

This chapter unites the object- and space-directed generalisation methods presented in Chapters 6 and 7. It presents applications of the problem-oriented workflow, focused on the comprehensive generalisation process introduced in Chapter 3. The aim is to highlight the modularity and flexibility of the approach taken to point data generalisation and the breadth of its applicability.

The first part illustrates the modularity of the overall generalisation process, while the second part presents its applicability through a series of maps generated from two different datasets. The maps are all screen captures of the development prototype. Appendix A provides an in-depth description of the prototype to the interested reader, including a feature description, screen shots and a high level class design.

9.1. Workflow overview

The generalisation process based on the problem-oriented workflow for point data generalisation introduced in Section 3.4 (Figure 3.3) represents a flexible and modular solution to point data generalisation. The process shows how point generalisation can be adapted in a modular way to data types, available attribute data and user interaction to generate the desired outcome, depending on the task and the questions a map user wants to address while using the map interactively. The process can be applied in a static way, applying predefined generalisation strategies (see 3.4.5) or in an interactive way, illustrated as in Figures 9.1 and 9.2.

From an interaction point of view, the map user, the context and the map purpose, shape the way in which the map is used interactively. One example might be a mobile user seeking a particular location in an unfamiliar city. Another might be a scientist, or data journalist, requiring a set of visualisations for a large spatial dataset in order to gain an insight of the data, and use the map as a visual analytics tool, with frequent user interactions.

Additionally, the type of dataset plays a role, as it facilitates the usage of specific algorithms. For instance, if a dataset only provides a list of coordinate pairs with no attributes, selection based on attribute values is not feasible. With a rich dataset containing numerical and textual attribute values, the possibilities for visualisation and generalisation are more extensive.

Figures 9.1 and 9.2 illustrate, for object- and space-directed point generalisation, the modular and interactive approach the generalisation process offers. Both figures show a

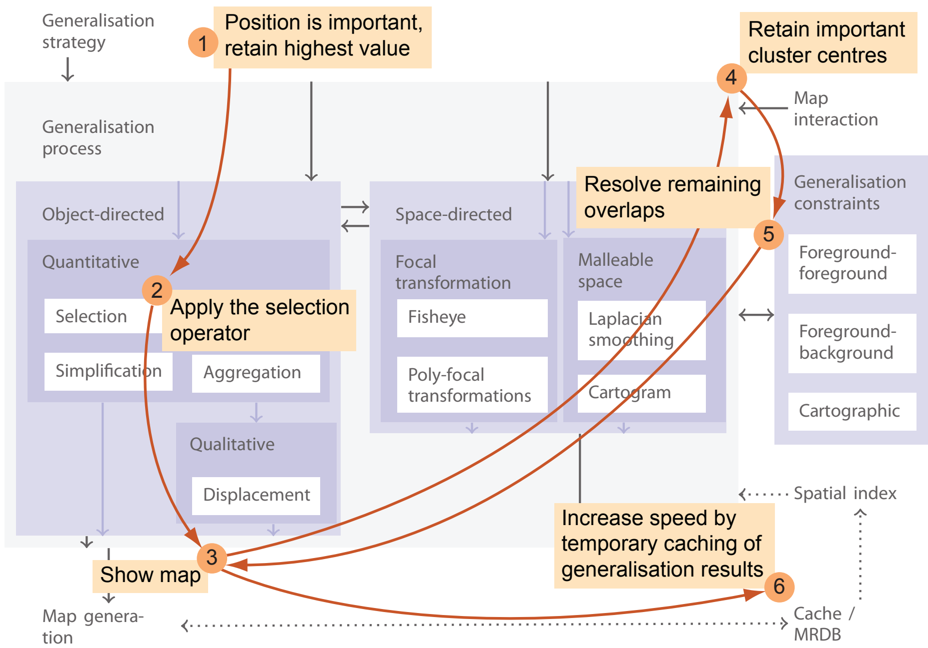


Figure 9.1.: Object-directed example of a walk through on the modular generalisation process (see 3.4). It shows the application of the point selection generalisation operator, where by user interaction parameters of the algorithm are changed and the caching is applied to further speed-up the zooming experience.

possible interaction sequence, and lead through it with numbered explanation boxes. The interaction sequence is overlaid on top of Figure 3.3 that shows the generalisation process.

An object-directed interaction sequence (Figure 9.1) might start (1) with a generalisation strategy stating that the position of the points is important, and the highest attribute values should be retained. In order to weight the attribute values and select the highest values, the next step (2) chooses the selection operator, and then a first map is generated by the generalisation process (3). Next, a user interaction requests that important cluster centres are to be retained (4) and emphasized, triggering a parameterisation change in the selection operator. Additionally remaining overlaps (across quadnode borders) are removed (5) and the map is updated. The application of a temporary cache (6) provides a smooth user experience for zooming and panning (see 6.5).

Figure 9.2 shows a mixed object- and space-directed generalisation sequence. In this workflow the position of the points, and the point pattern should be retained (1). Hence, in order to maintain the point pattern, simplification as a generalisation operator is selected (2). In order to enlarge dense regions and to provide further details, cartogram as a space-directed generalisation method is applied (3). In this workflow the user decides to remove

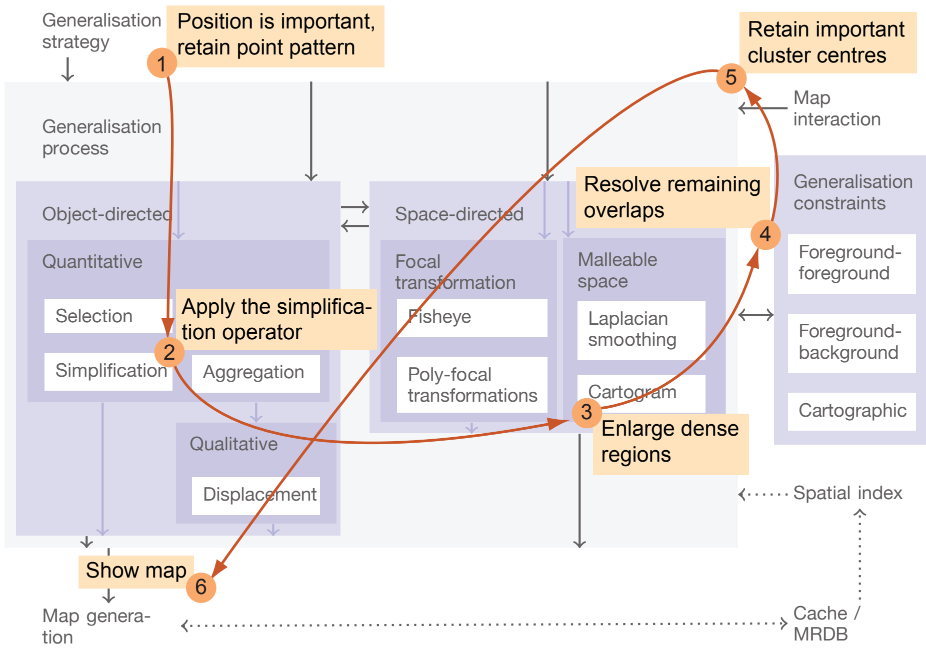


Figure 9.2.: Object- and space-directed example of a walk through on the modular generalisation process (see 3.4). The user first applies a simplification operator and then selects space-directed generalisation to emphasises dense regions, and filters out cluster centres.

remaining overlaps (4), and to retain and emphasize important cluster centres (5), and to generate the map only at the very end (6). Note, however, that map generation, would be possible after each described step.

The presented interaction sequences exemplify the flexibility of the generalisation process. They illustrate how the quadtree acts as the underlying spatial-index, that facilitates the modularity of the workflow. The next two sections present different map visualisations based on two sample datasets to highlight the variability of the map visualisations possible.

9.2. Use case: Swiss leisure activity dataset

The different map generalisation results presented in this section derive from the Swiss activity dataset (see Venkateswaran 2015 and Appendix B). The dataset stores, for each populated place, a frequency list of activity terms attributed to geocoded photographs retrieved from Flickr – a web service to share pictures – in that place.

The dataset covers, in addition to numerical attributes like the occurrence of the most frequent activity, textual attributes such as the names of the most frequent activity. The national coverage of the data, as well as the textual and numerical attributes, provide an ideal case study to apply the generalisation algorithm and present different map results.

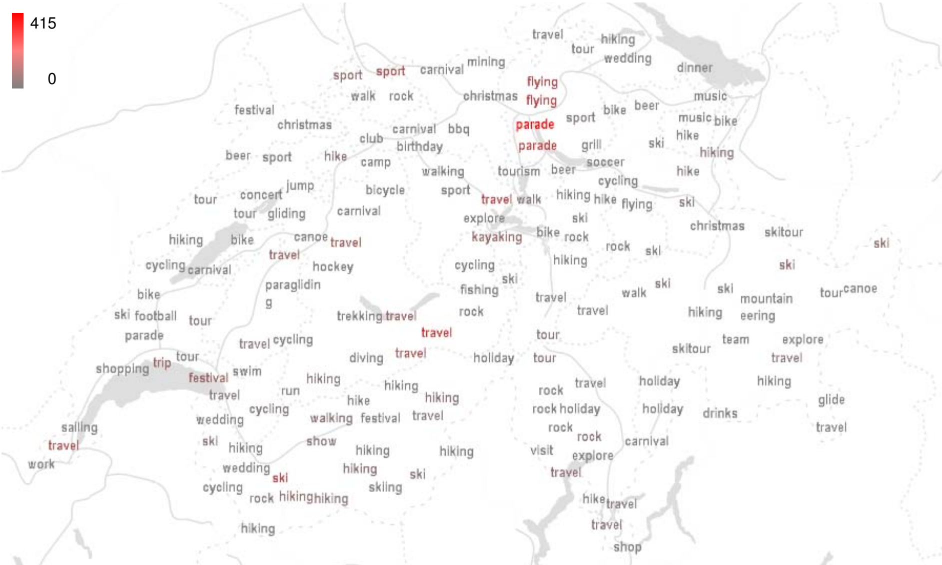


Figure 9.3.: Most frequent activity per populated place. The number of times the activity occurs is shown in shades of red (higher values show darker on a logarithmic color scale). Base map: Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL

The following four maps (Figures 9.3 to 9.6) depict different generalisation results derived from the activity dataset. These maps also include generalisation results, where quadtree-based generalisation is applied to labels. Three maps present the generalised activity labels (Figures 9.3,9.5,9.6), instead of icons or other symbolisation, which help to see if the quadtree-based generalisation may also be applied to labels. It provides an insight into whether the method is applicable to labels, and its limits.

As a use case, we envision a researcher who wants to get a first impression of the dataset and seeks an overview on how different activities are distributed in Switzerland. At first the distribution of the most frequent labels is of interest. Figure 9.3 presents a generalised map of the most frequent activities for each populated place in Switzerland. The number of activity tags per place are represented on a colour scale from grey to red. It shows that activities within highly populated places stand out, such as the activity *parade* in Zurich and *travel* in Geneva or Lucerne. High numbers of the most frequent activity term also occur in touristy regions such as *travel* in the Bernese Oberland or *ski* in the Rhone valley.

In contrast to map symbols – such as circles, squares or small icons – map labels occupy more space on a map. In terms of readability overlapping labels are even less desirable than overlapping symbols. Hence, a map with a generalised set of labels has a lower capacity to represent point data. For comparison, the map in Figure 9.4 depicts 2.5 times more point symbols (squares) than the labels shown in the map of Figure 9.3.

If quadtree-based generalisation is used to place labels, care has to be taken that the

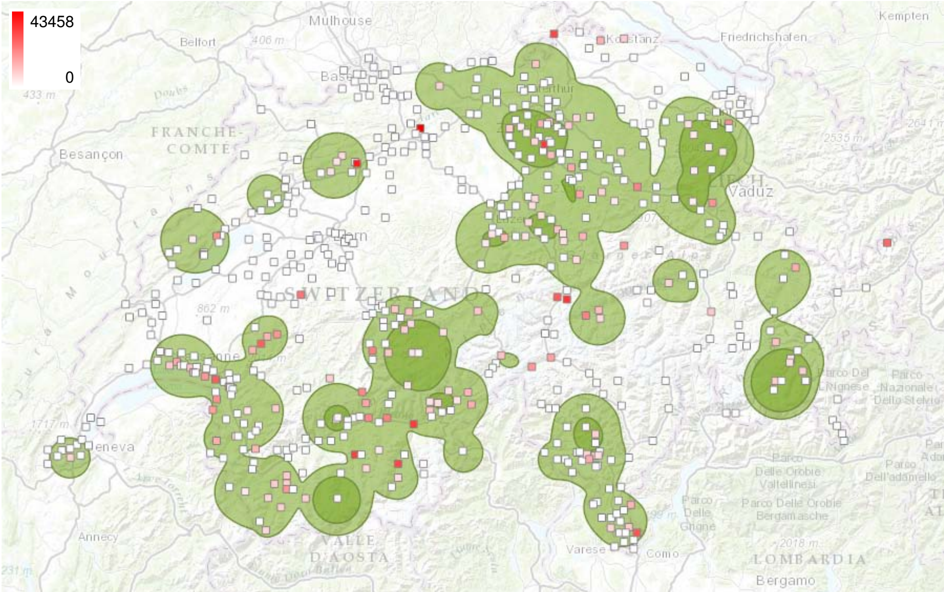


Figure 9.4.: Number of tags per populated place in shades of red (higher numbers show darker on a logarithmic color scale) compared to the 50 % and 95 % KDE contour of the terms *hike* and *hiking* (green) Base map: ©2013 ESRI ArcGIS Services World Topo Map

width of the label does not exceed more than twice the smallest quadnode width at the displayed map zoom level. Long labels such as *mountaineering*, should be hyphenated, to better approach the shape of a square. Alternatively, the algorithm can be extended to check for overlap with all the points in the quadnodes that the label covers, not only the immediate neighbours. Figure 9.3, 9.5 and 9.6 show quadtree-based selection applied to labels. Here, the labels are not further displaced, which would be needed to create more room to place additional labels.

Figure 9.4 and the following two Figures 9.5 and 9.6 present how point data generalisation of foreground features is applicable in combination with additional data layers, such as KDE analysis, with the background map serving as a spatial reference.

The next map (Figure 9.4) presents the distribution of the overall number of activity tags per populated (in shades of red), and compares it to the contour of the KDE 50 % and 95 % contours of the activity terms *hike* and *hiking*. The foreground objects, on this map, are point symbols showing the number of activity tags per populated, slightly generalised by quadtree-based selection, in order to guarantee that the places with the highest counts are kept. In this example slight overlaps are tolerated to retain as many places as possible.

If multiple map content layers are combined, generalisation algorithms should optimally include background constraints (see Section 6.7). However in this example, a reasonable selection of symbol or label size, combined with a higher degree of generalisation, facilitates the combination of several map layers to generate a readable and interpretable

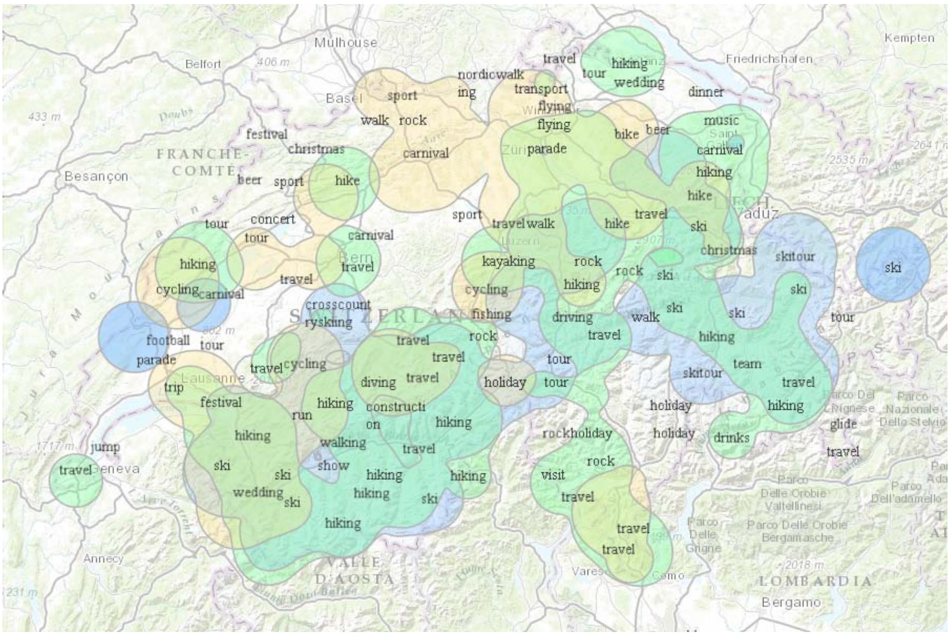


Figure 9.5.: Most frequent activity term per populated place compared to the 95 % KDE contour of the activities *ski* (blue), *cycle* (yellow) and *hike* (green). Base map: ©2013 ESRI ArcGIS Services World Topo Map

map. In these examples the map data layers are both derived from the same dataset and provide further insight into datasets.

Figures 9.5 and 9.6 both show in the foreground the generalised most frequent activity term per populated place[1]. The background represents different contours of the KDE 95 % contour for three activity terms in Figure 9.5 and two in Figure 9.6. These figures present a background carrying more information and overlap than the previous two maps. In these examples it is the 95 % KDE contours of selected activity terms carrying the main information layer of the map, while the foreground serves rather as a reference to lookup what the important terms are in the region of interest. Hence the foreground labels should hardly clutter. This can be observed especially in Figure 9.5, where the contours for *skiing* in blue, *hiking* in green and *cycling* in yellow, all overlap. As an aside, it is interesting to note that these activities clearly have different spatial extents. Similar observations hold for Figure 9.6 with the pink contour line representing the activity *carnival* and the violet one representing the contour of the term *festival*.

[1] An alternative method to retrieve the most particular activity per location is to apply the *term frequency – inverse term frequency* (tf-idf) method. A well known method in data mining that provides the means to rank the activity terms in order to apply map generalisation methods.

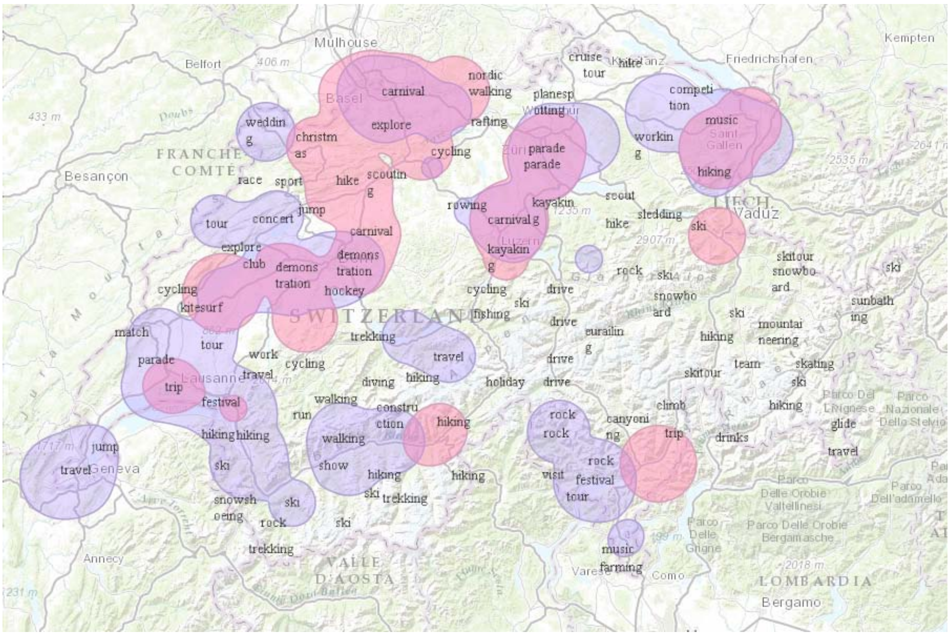


Figure 9.6.: Most frequent activity term per populated place compared to the 95 % KDE contour of the activities *carnival* (pink) and *festival* (purple). Base map: ©2013 ESRI ArcGIS Services World Topo Map

9.3. Use case: Earthquake dataset

Besides other parameters the ECOS-09 earthquake catalogue contains the moment magnitudes (MW), coordinates of the epicentre of measured seismic events, earthquakes and other induced seismic activities in Switzerland and the bordering regions.

The following visualisations of the dataset illustrate the likely results required by a data journalist, who may seek to investigate an earthquake dataset in order to gain further insights on the data and as a means of illustrating these insights.

The visualisation of over 6400 registered seismic events, even with tiny symbol sizes, may not convey the message; of where most seismic events cluster, as strongly as can be done with a centroid-based aggregation and scaled symbols. In Figure 9.7 the scaled symbols represent the number of registered activity at the centroid point of the aggregated events. This type of visualisation nicely highlights the Northern side of the Rhone valley, less intensely the South Eastern part of the Grisons and in the North an earthquake cluster in the Canton of Basel.

From knowing where the earthquakes cluster in a specific region, one might then want to further investigate whether the (non-logarithmic) magnitudes cluster, and see if low or high values cluster in space. *Getis-Ord G** is a local geostatistical method (O'Sullivan and Unwin, 2010) that represents for each event whether the values are concentrated for

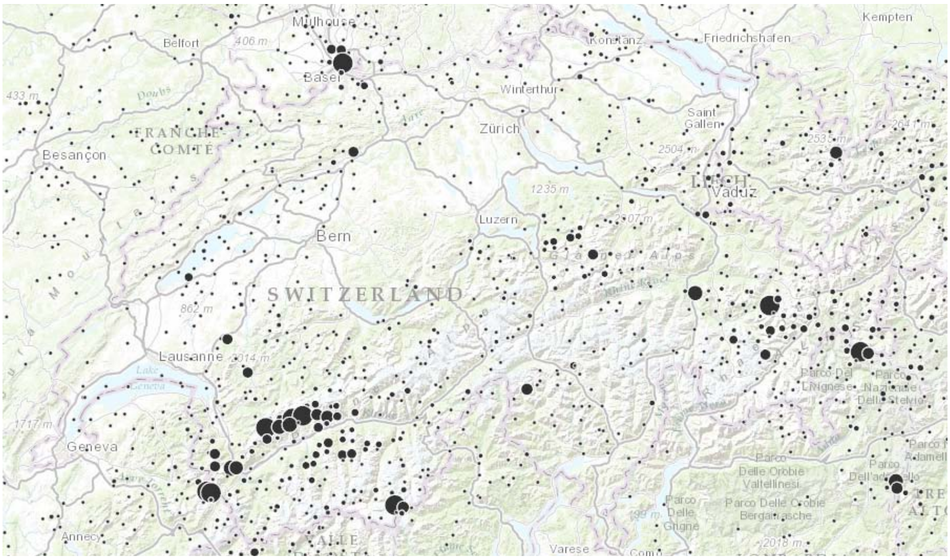


Figure 9.7.: Distribution of earthquakes in Switzerland 2001 – 2008, generalised by the centroid based aggregation with graduated symbols representing the number of registered earthquakes. Base map: ©2013 ESRI ArcGIS Services World Topo Map

low or high values. Figure 9.8 presents this statistic for the magnitude of the registered seismic events[2]. The results of the local statistics are generalised with a quadtree-based selection operator to reduce the clutter and to maintain the spatial point distribution. Clear clusters, such as in Figure 9.7, however, do not emerge.

For a data journalist, another interesting map might then be a space-directed map visualisation (see: Chapter 7) such as a map that enlarges those regions of Switzerland with a high occurence of seismic events, as identified in Figure 9.7. Figure 9.9 shows such a map; it depicts the quadtree-based cartogram for the moment magnitudes of registered earthquakes in Switzerland. For reference as the foreground map, all registered events are represented as small red dots.

An alternative representation is a quadtree-based point selection of the highest magnitude values, overlaid on the weighted KDE of the measured magnitudes as shown in the last map of this series, Figure 9.10.

While both maps carry the same main message, highlighting where most seismic events cluster, a data journalist needs to carefully think about which map conveys the message best.

[2]Getis-Ord G* is applied to the non-logarithmic scale of the moment magnitude.

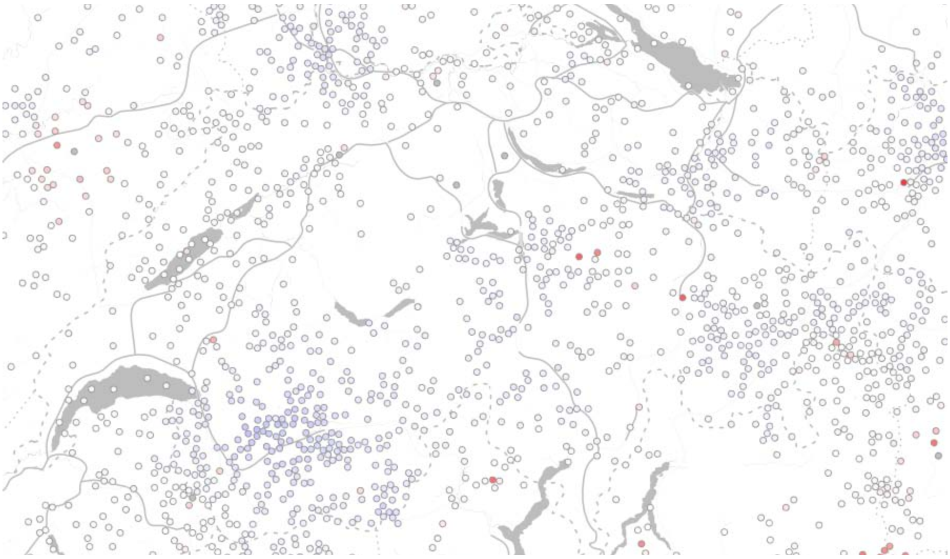


Figure 9.8.: Distribution of earthquakes in Switzerland 2001 – 2008 b) Getis-Ord G* statistic of the earthquake's moment magnitude *mw* – selection by maximum value. Base map: Stamen Design, under CC BY 3.0. Data by OpenStreetMap, under ODbL

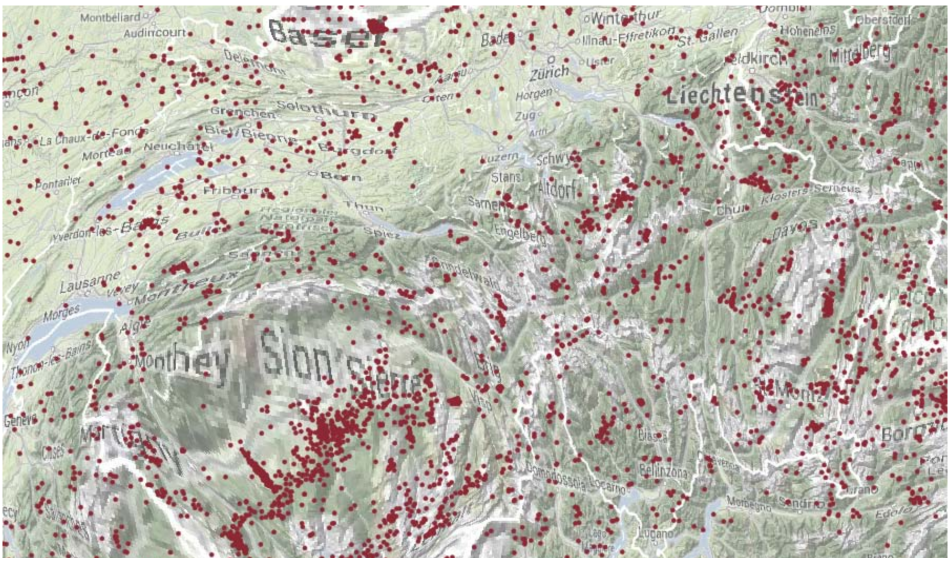


Figure 9.9.: Distribution of earthquakes in Switzerland 2001 – 2008, generated by a cartogram. Base map: ⓒ2013 Google Maps

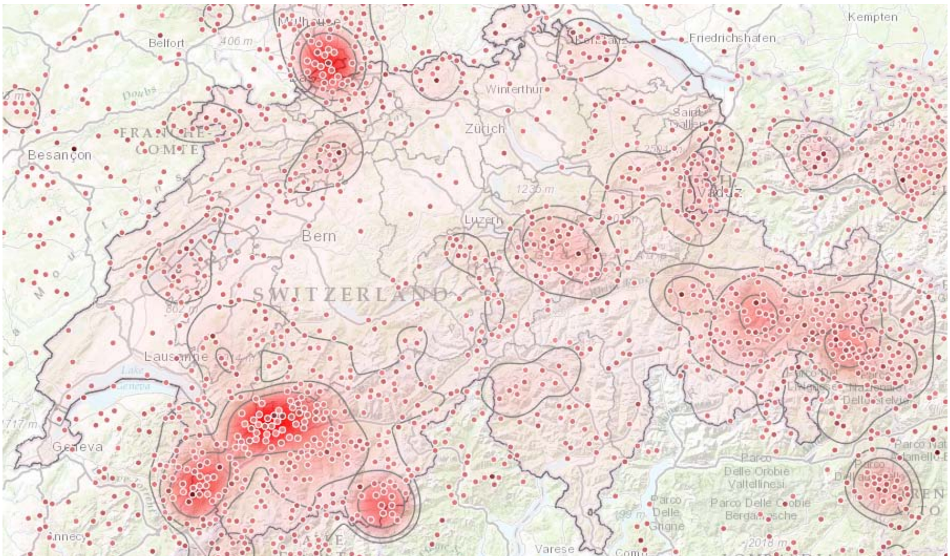


Figure 9.10.: Distribution of earthquakes in Switzerland 2001 – 2008. Quadtree-based selection applied to the earthquake's moment magnitude overlaid on weighted KDE of the moment magnitude *MW* – selection by maximum value. Base map: ©2013 ESRI ArcGIS Services World Topo Map

9.4. Concluding remarks

This chapter demonstrated the modularity and the applicability of the overall workflow of point data generalisation, based on two case studies and datasets. It further illustrated how the quadtree-based generalisation can be applied to points symbolised by labels. It showed how special applications, such as geo-statistical or data mining methods, can be integrated or combined with map generalisation. The chapter provided examples of maps that contain several information layers, where the generalised point data are shown in combination with a second information layer such as a contour layers (Figures 9.3, 9.5, 9.6).

Chapter 10.

Discussion

This thesis aimed to develop concepts and methods for real-time point generalisation, with a particular focus on mapping for portrayal in mobile and web-based geographic information systems.

This work responds to the need in interactive mapping environments to provide a framework of algorithms to address the recent paradigm shift afforded by information science. With the advent of various data providers facilitating the 'mash-up' of geographic data, point data has become the most common geometric data type. Unlike line generalisation, however, point data generalisation has received limited attention in research so far. This work therefore presents, on the basis of the quadtree data structure, a modular framework for point data generalisation to address the particular requirements in web and mobile mapping. The framework is more comprehensive than any other solution that has been presented in the literature so far, including all relevant generalisation operators for point data.

This chapter present an in-depth discussion of the methods proposed and the results reported in this thesis. Section 10.1 focuses on the discussion of the quadtree-based, object-directed generalisation algorithms proposed, while Section 10.2 reflects on the results obtained from the cartographic analysis of these object-directed algorithms. Section 10.3 discusses the methods that were proposed for space-directed generalisation, as well as the results of the experiments that were carried out with these algorithms. Section 10.4 addresses the overall proposed methodology and workflow of quadtree-based point generalisation, as well as the specific interaction technique of content zooming. In each of these four sections, the contributions as well as the insights and limitations of the corresponding results are identified. Section 10.5, then, revisits the research questions that have been guiding this work, as laid out in the introductory chapter. The chapter is concluded in Section 10.6 by a set of recommendations that can be derived regarding the usage of the methods that have been proposed in this work.

10.1. Object-directed point generalisation: Methodology

Object-directed generalisation algorithms manipulate map objects or fields depicted on a map, through the manipulation of vector or raster data. In other words, these are map generalisation operations in the classic sense of cartographic map generalisation. The overall framework introduced in Chapter 3 presents a methodology for point map generalisation and defines object-directed generalisation, as well as relevant generalisation operators.

10.1.1. Contributions

Chapter 6 proposes a comprehensive set of algorithms that implement the major generalisation operators for point data outlined in the methodology of Chapter 3, with a focus on real-time generalisation. The very characteristic nature of the quadtree data structure serves as a basis for enabling the definition of these algorithms, due to its density-based, hierarchical index structure.

The use of the quadtree allows bridging the two main approaches reported in the literature (Weibel and Burghardt, 2008; van Oosterom, 2005; van Oosterom and Meijers, 2011), which have been tackling real-time generalisation either by pre-computation using hierarchical data structures, or by efficient, simple heuristics that allow a certain degree of flexibility.

The advantage of using hierarchical data structures lies in the fast retrieval of a pre-computed generalisation solution. The advantage of using heuristics lies the possibility to design algorithms that are fast (yet simple) and allows flexibility. The disadvantages are non-adaptive and inflexible data structures in the first case (pre-computation), while solutions in the second group sacrifice cartographic quality to reduce computational complexity.

The object-directed, quadtree-based generalisation algorithms implement the major generalisation operators for point data (Edwardes et al., 2005) through the quantitative point reduction operators, *selection*, *simplification*, and *aggregation*, and through *displacement* as a qualitative point displacement operator. Our framework presents, for each of them, one or more algorithms with different aspects. Additionally, it integrates methods to mitigate quadnode border conflicts to remove remaining overlaps, if required, as well as solutions to reduce homogenisation effects at small scales.

10.1.2. Insights and limitations

Working principle The working principle used by the generalisation operator based on the quadtree can be summarised as follows: data-driven spatial subdivision; local and topological analysis of the spatial point configuration; execution of the selected generalisation operator; and refinement based on the resolution of generalisation constraints. Hence, the approach inherently follows a common approach in cartographic map generalisation, which is *measurement*, *characterisation* and *evaluation* (AGENT, 1998; see also McMaster and Shea, 1992, Brassel and Weibel, 1988, Harrie and Weibel, 2007).

Data-driven and hierarchical subdivision A main property of the point region quadtree is, that the data structure is data-driven and allows for incremental updates (Samet, 2006). It hardly creates an overhead of the data, that is, the index itself takes little additional storage in order to perform queries on the index. The data-driven subdivision and the hierarchical structure facilitate queries on the local neighbourhood of the points, as well as their local density.

The approach does not necessitate laws defining the number of points at the target map scale such as are often applied in combination with the Radical Law (Töpfer and

Pillewizer, 1966). Mesh-simplification (Burghardt and Cecconi, 2007) – a point reduction algorithm relying on the radical law – may not resolve all conflict constraints, especially for strongly clustered patterns, as the Radical Law does not consider the spatial configuration of the points (cf. Section 8.4).

Modularity and adaptation A main strength of using the quadtree as an index structure and then applying the generalisation operators on-the-fly on the quadnodes linked to the desired zoom level, is the resulting modularity and flexibility of the approach. The quadtree index is built once and the generalisation operators, constraints and parameterisations can be changed without rebuilding the overall tree. In addition to the real-time performance of the generalisation operation, this approach enables an adaptive modification of operators *e.g.* based on scale or context. Furthermore, additional generalisation concepts or filters that can be integrated to a local operation could be easily added. For zoom levels that lie in between the zoom levels given by quadtree level structure, the approach would allow for interpolation methods (Cecconi and Galanda, 2002; Cecconi, 2003) to enable a continuous zooming experience. The quadtree-based generalisation algorithms would then also include transition rules between the 'main' base-2 zoom levels.

The presented approach is restricted to static data, as dynamic data add an additional degree of complexity to the overall generalisation, which would have gone beyond the scope of this thesis. The inclusion of dynamic data would additionally require the definition of temporal generalisation operators and concepts.

Performance and caching Computational performance that enables real-time behaviour is key in an interactive map environment. Section 8.2 presents and discusses the performance figures of the approach based on three datasets of different sizes, without the use of caching, that is, generated on-the-fly. As the quadtree is data-driven, performance is slightly influenced by the spatial distribution of the point data.

The creation of the tree comes with a higher performance cost, as opposed to the average computation time needed to generalize between two different zoom-levels. Of the generalisation operators, not surprisingly, displacement has the highest performance cost, due to topological queries. Additionally, operations involving quadnode neighbor queries – such as border conflict resolution – come with a slight increase in computation time (see also Section 8.5). Nevertheless, in general it can be concluded that the proposed algorithms have demonstrated real-time performance that are clearly in the sub-second range even for large datasets of several tens of thousands of points.

For massive datasets the creation of the index can be off-loaded to the data provider (i.e. computed server-side) or directly processed while importing the dataset. As the index is supported by all quadtree-based generalisation operators, switching the operators or changing the parameters does not require a rebuilding the index.

Caching as described in Section 6.5 is an additional solution to significantly reduce the computation time between the zoom levels for massive datasets. The drawback of a dynamical cache is that it requires an updating of the cache if operators, or their parameterisation, are changed.

While the quadtree approach potentially handles very large datasets, the prototype application has some limits as it stores the complete dataset in the main memory. Furthermore, moving the application from alpha to beta stage probably would improve the upper bound of data points the application can presently handle.

Parameters While the quadtree data structure is data-driven, the quadtree-based generalisation approach also relies on parameters that have an influence on the generalised map results. Quadtree parameters affecting the generalisation results are the position and spatial extent of the quadtree root node, and the basic symbol size.

The position and extent of the quadtree affects the spatial subdivision. It influences which points fall inside a quadnode and are processed together for each zoom level. Hence, point clusters with an equal number of points are not processed in the same way depending on their position. One point cluster may fall inside a quadnode whilst the other is processed in up to four neighbouring quadnodes. Artificial visual groups form if individual points are positioned close to the quadnode border. Both described subdivision states are intrinsic to a regular spatial subdivision, where the spatial division does not inherently consider spatial clusters. A symbol-to-node ratio close to 1, considering neighbour nodes during the generalisation, the use of graduated symbols, or the use of constraint rules, reduce this effect.

Grid-based tile map services belong to one of the most widely supported services on the web and integrate well with quadtree-based generalisation approaches. An alternative concept is the Quaternary Triangular Mesh QTM (Dutton, 1999a,b), providing a global regular tessellation and spatial index structure across the globe. QTM is closely linked to the quadtree, as it also uses a subdivision into four child nodes, though of triangular shape (equilateral triangles). The work by Dutton (1999b) proposed cartographic line generalisation algorithms. The adaptation of the presented quadtree-based methods into the QTM possibly is an interesting further application integration of the quadtree-based generalisation.

Symbol-to-node ratio defines the ratio of the default symbol size and the quadnode width. A ratio close to 1 means that the symbol size is about the size of the smallest quadnode displayed at the current map zoom level. Thus, a ratio close to 1 results in a rather dense generalisation result, while a ratio around 0.2 is comparatively sparse. Changing the symbol-to-node ratio in the generalisation process requires either a change of the quadtree bounding box, and therefore the re-build of the point region quadtree, or an adaptation of the symbol size.

Constraints Section 3.1.3 presented the various constraints on point data that can be formulated for point data. These are: foreground-foreground constraints, cartographic constraints and background constraints (Table 10.1). Constraints can be seen as a limitation that reduces the number of acceptable solutions to a problem (Weibel and Dutton, 1999).

Foreground-foreground constraints are inherently maintained by the data-driven nature of the quadtree. That is, depending on the selected operator proximal and topological

Table 10.1.: Overview of how quadtree-based generalisation operators address point data constraints

Constraint	Description
Foreground-foreground constraints	Symbol-to-node ratio, grouping constraints
Cartographic constraints	Border conflict constraints, minimum distance threshold, symbol size
Background constraints	Conceptual solution presented for raster and vector background constraints

relations persist over the generalised map scales. The foreground-foreground constraints are influenced by the symbol-to-node ratio, the grouping constraints and the density constraints. The grouping constraints emphasize dense clusters over the overall coverage of the point pattern. Grouping constraints affect the macro level as they control the thinning process, based on the depth and the density of the quadnode. Depending on the map purpose and the user's point of view, the desired amount of thinning is different especially for strongly generalised point datasets at small scale zoom levels. The small participant experiment on grouping constraint (Appendix C) shows high agreement on the least liked solutions, but when it comes to the preferred solution there is less agreement at the lowest zoom level compared to the degree of agreement at the highest zoom level. This also showed in the additional comments provided by the participants.

Cartographic constraints are covered by the minimum distance threshold applied in the conflict constraints, and the setting of minimum and maximum symbol sizes. The distance threshold sets the distance or the amount of overlap between map symbols. The border conflict constraints then fully resolve all the remaining conflicts occurring at quadnode borders. The setting of the symbol size that remains the same throughout the scales ensures the perceptibility of the point symbols.

Background constraints were covered by a conceptual solution in this thesis (see Section 6.7), which proposes how raster and vector background constraints can be addressed in the context of quadtree-based algorithms, in order to retain second order spatial effects in the dataset (O'Sullivan and Unwin, 2010). The approach requires the background data to be in a similar scale range as the target scale, especially in the case of vector data. The limits of the proposed approach may be tested with a proper working extension of the current implementation of the prototype of this thesis.

Reasonable constraint formulation relies on an applicable scale level and some sort of 'knowledge' of which features constrain the foreground points, which is not known *a priori*, and needs to be formulated. This requires that the semantics of geographic features on the background layer are known. This holds for background and foreground data. The encoding of a line feature as a river, or specific pixel values as water bodies, allows the formulation of background constraints only if the semantics of the foreground data is specified as well. Meaning, it is known whether a point symbolises a restaurant or a bird observation.

The conceptual solution of background constraints for quadtree-based object-directed generalisation assumes that the semantics of the data are provided to the system. In a real-world environment foreground and background data rarely come 'semantically' encoded in a standard way, that would allow the formulation of automatic and generic semantic background constraints. These prerequisites are hardly given, especially if the system allows for generic background data and allows for any type of point data. Therefore, in map generalisation there is a need for methods that automatically interpret heterogeneous data (Sester et al., 2014) from multiple sources, and this need is also due to the increased availability of various geographic data sources.

Even if foreground and background data comes in a standardised way, it highly depends on the understanding of how the background data influence, and hence constrain, the foreground data. A lack of understanding of how the background data constrain foreground data is likely to lead to logical inconsistencies on a map. This ranges from POI being moved inside a lake, or to the other side of a road, to the point of generalising accident data, where it is critical that the relation between road intersection and accident is maintained, and that the data is aggregated along, rather than across, the road segments.

Generalisation operators The implemented operators are: *selection*, *simplification*, *aggregation* and *displacement*. The operators are represented by one or more quadtree-based algorithms, summarised in Table 10.2. Each of the proposed algorithms has its own scope of applicability, depending on the data type and available attributes as well as the generalisation process selected and provided interaction possibilities available. The presented algorithms are a subset of likely point generalisation algorithms that are feasible using the quadtree as an index structure. All implemented algorithms generalise the map locally or at the micro level, according to the terminology used in the AGENT project (Barrault et al., 2001).

Value-based selection based on the value of point attributes enables the use of many different local selection functions. Retaining local minima and maxima are probably the most straightforward functions. Further applicable functions are *ranking* based on a ranking model including multiple attributes (e.g. opening times and popularity of a POI), frequency of occurrence, or the application of local descriptive statistics, or local autocorrelation statistics. Depending on the point feature type, in case of associated texts, descriptions or tags, measures originating from information retrieval such as *tf-idf* (short for 'term frequency-inverse document frequency'), *term frequencies* (see Figure 9.5) and others provide additional selection criteria.

Care has to be taken if several points in a quadnode all equally score the highest attribute value. In this case it is best to either choose a random point, or select one, based on spatial criteria. This avoids the generation of a seemingly uniform pattern, particularly if the scale of measurement is ordinal. Additionally, in order to underpin the selection criteria visually, the use of a colour or symbol size scale highlighting the points accordingly provides the user with additional context and a better understanding of the generalised map.

Centrality-based simplification, *quadnode center aggregation*, *midpoint aggregation*

Table 10.2.: Overview of quadtree-based, object-directed generalisation operatores and the associated generalisation algorithms

Operator	Algorithm	Description
Selection	Value-based selection	Returns the most relevant point per quadnode based on an attribute-based selection function.
Simplification	Centrality-based simplification	Retains the most central point per quadnode; no attribute data required.
	Weighted centrality-based simplification	Extends centrality-based simplification by using attributes as weights (variant: inclusion of neighbour node weights).
Aggregation	Quadnode centre aggregation	Aggregation of points to the tile centre (variant: with density-based, graduated symbol sizes).
	Midpoint aggregation	Aggregation of points to the mean centre. Variants are, density-based, graduated symbol sizes and inclusion of neighbour node weights.
	Cluster-based aggregation	Aggregation based on clustering criteria and quadnode neighbours.
	Collocation filtering	Aggregates points based on collocation rules inside nodes, or as a variant by including quadnode. neighbours
Displacement	Displacement	Locally displaces points of a node to empty neigboring nodes, analysing either only the direct neighbours, or also the diagonal neighbour quadnodes.

and *displacement* are all based solely on the point's location, as a required input to the generalisation algorithm. Displacement, however, relies on the preceding point reduction operators that may be based on point attribute values as well.

Of these algorithms (Table 10.2) *midpoint aggregation* is the most basic one and the one that homogenises the point pattern the most. In certain cases, it might neverteless be graphically appealing, applying small symbol sizes, as it shares similarities with the half-tone printing process. From a cartographic perspective, *midpoint aggregation*, however, is less appealing. *Quadnode centre aggregation*, similar to the approach by Burghardt et al. (2004b) and *Cluster-based aggregation*, do not give the impression of a homogenised outcome.

Co-location filtering is a special case, compared to the other proposed algorithms. It relies on the co-location of at least two categories of point features within a specified area.

This implies that, the larger the area to search for co-located features, the more likely it is to find them. Hence, the zooming experience with collocation filtering is somewhat opposite to 'normal' generalisation, as the more the user zooms in, the less likely map features are to be displayed. The inclusion of the local quadnode neighbourhood mitigates the effect of ignored co-locations across quadnode boundaries.

Displacement as an operator is limited to small scale changes. Depending on the underlying point pattern and and the data reduction rate (see Section 8.4), the displacement algorithm is able to retain up to around 20 % more points on the map. As these points typically enlarge and 'explode' cluster centres (Figure 8.11b on page 122), the generalised point pattern has a tendency to homogenise earlier in the overall zoom chain (Section 8.9). Compared to the iterative displacement operator by Mackaness and Purves (2001) and the icon labelling algorithm by Harrie et al. (2004), due to the use of the quadtree as an index structure, the quadtree-based displacement algorithm benefits in terms of performance. No iterative search is needed to search for empty space. The integration of these two working principles for displacement (Harrie et al., 2004; Mackaness and Purves, 2001) – especially the concept of least-disturbing space for label placement (Harrie et al., 2004) – into the quadtree-based solution would be an interesting extension of the family of quadtree-based point generalisation algorithms.

Common to all basic quadtree-based algorithms is that the subdivision into nodes generalises all elements from the node's subtree into one representative point. This implies that one point may represent a large collection of points, while points across different quadnodes may visually appear as a group but represent a much smaller number of points. The application of graduated symbols or the extension of the border conflict constraints with an artificial group distance threshold avoids the creation of artificial visual groups. The inclusion of the topology, and the neighbouring quadnodes in the basic quadtree algorithm, improves the cartographic quality, but also increases the processing time.

While McMaster and Shea (1992) consider symbolisation as a generalisation operator, this work assumes that symbolisation is defined at the beginning of the generalisation process and is defined by the user or the map application, being principally a map control rather than a generalisation operator, in the same way as in Brassel and Weibel (1988) and AGENT (1999). While it is important for semantic integration, *a priori* reclassification and symbolisation of data with unknown semantics is hard to solve in a satisfactory way.

10.2. Object-directed point generalisation: Cartographic analysis

Compared to traditional maps, the cartographic quality of digital maps is often more relaxed due to the more flexible and modular approach during map creation. That is, there is a trade-off between cartographic quality and computational costs, even more so in a real-time environment.

Cartographic map generalisation is an inherently subjective process, as is, ultimately, its evaluation. One possible way of assessing the quality of the generalised map and comparing generalisation solutions is through evaluation by cartographic experts. Criteria applied between experts may differ significantly, the issue of including more objectivity in the evaluation process has been addressed by several authors (Stoter et al., 2009; Bard, 2004a), see also Section 5.1.1.

The cartographic quality of the quadtree-based, object-directed algorithms overall, and the generalisation operators, is addressed in Chapter 8. The evaluation applies a quantitative approach to assess the strengths and the weaknesses of the presented algorithms. The analysis investigates key aspects of map generalisation: the aspects of data reduction, conflict reduction, data enhancement, displacement measures, density variation, homogenisation and cluster maintenance. Depending on the investigated aspect, different point datasets, differing in type (PColl, POI), volume and geographic extent, serve as a basis and case study for the analysis.

Data reduction The data reduction of the quadtree-based algorithms reflects the data-driven nature of the quadtree. Compared to the Radical Law (Töpfer and Pillewizer, 1966), the smallest cross-section between both curves (see Figure 8.2) shows the difference in the resulting generalisation outcome.

The highest reduction rate for quadtree-based algorithms indicates the zoom levels where most cartographic conflicts occur. At these zoom levels the Radical Law retains less points than the quadtree-based algorithms, whereas it retains more points at small zoom levels. For small zoom levels algorithms, strictly following the Radical Law results in a comparatively more cluttered outcome (see Section 8.4).

Quadtree-based algorithms retain comparatively more points for intermediate zoom levels. This leaves room for further reduction or change of generalisation parameterisation, if the point pattern is to retain a clustered characteristic.

Cartographic conflict reduction Aside from data reduction, an important aspect of map generalisation is the evaluation of how legibility constraints are maintained. The analysis of the conflict-reducing capabilities of the quadtree-based generalisation was presented in Section 8.5. Depending on the application and the desired aesthetic quality of the map, a certain degree of cartographic conflicts – that is, of overlap between neighbouring point symbols – are tolerated. In the basic application of quadtree-based generalisation not all cartographic conflicts are resolved, due to remaining overlaps at the border of the quadnodes. The application of border conflict resolution strategies, by selection or small displacement, fully removes these remaining conflicts.

Two aspects need to be considered. First the additional processing cost due to neighbour lookup (see Figure 8.7). The additional processing cost can be reduced by directly storing in each quadnode a reference to its neighbouring nodes. Another option is to move all point features inside the quadnode's boundary. This solution is less desirable from a cartographic point of view, however, as it introduces a certain regularity to the generalised point pattern.

The second aspect is to choose of a border conflict resolution strategy that reflects the nature of the chosen generalisation operator. Thus, for instance, for point reduction operators and a map purpose where the displacement of point is suboptimal, border conflict solution *avoid* reflects the nature of point reduction better than *move*.

Data enhancement One key goal of map generalisation is not only to retain, but also to highlight and exaggerate essential characteristics of the geographic map data (McMaster, 1987). Section 8.6 studied how quadtree-based local selection compares to a global value-based selection operator by retaining the same number of points. The comparison illustrates how the quadtree-based local selection retains the local point context of the point pattern. It shows that the quadtree-based solution better preserves the local context of the data, by retaining the values that are locally most important. In comparison to the global selection, however, global maxima are presented less clearly, due to the retained point context.

The quadtree-based approach provides a means to visually enhance areas of high point densities. This is achieved by applying clustering or density criteria and retaining only the content of those nodes satisfying the chosen criteria. The decision of which selection operator is chosen depends on the application context. That is, whether more importance is given to the global maxima of the point distribution or the local one or, whether the generalisation outcome should strive for a compromise between both.

Quadtree-based co-location filtering only retains selected categories of map features that are in the vicinity of each other. Co-location filtering differs from the other operators as the selection operation is based on a combinatorial rule. Furthermore, co-location in a certain sense inverts the zooming experience. The higher the map zoom level, the less likely it is to find co-located map features, as the quadnode area to which the co-location rule is applied is reduced. In that sense, selection based on a co-location rule is predominantly a visual exploration tool, which allows the exploration of a topological aspect over a range of scales, rather than a generalisation operator in the classical sense.

Displacement Apart from the point reduction operators, the quadtree-based approach also offers a displacement operator. Displacement, as a qualitative generalisation operator, works locally and is often used as the last operator in the chain of generalisation steps, operating in a narrow band of scales for a given number of point features. Additionally, a premise of displacement is to be able to create sufficient free space within a point pattern. Displacement increases the cartographic quality of the generalisation, and improves the generalisation solution of the point reduction algorithms with a good overall performance (see Section 8.2).

Section 8.7 compares the displacement capabilities of the proposed algorithm and the effect on the density distribution of the point pattern, and evaluates the cummulative displacement vectors. A comparison between simplification and displacement operators demonstrates the additional number of points that can be accommodated by the displacement algorithm.

Depending on the point distribution and the data reduction rate (see Figure 8.1 on page 113) the displacement algorithm is capable of displaying a substantial amount of additional points on the map, in the range of up to about 20 % between two consecutive zoom levels. The algorithm places the additional point features typically along the border of existing clusters, which is reflected in the density distributions of the point patterns. Thus, partly given by the nature of the displacement operator, the quadtree-based displacement operator relatively decreases the density of the point pattern at the cluster centres, and increases the density at the border of the clusters.

The analysis of the displacement vectors reflects the working principle of the algorithm. That is, if displacement is only applied to the horizontal and vertical neighbouring nodes, the cumulative sum of the displacement vectors classified into 16 cardinal directions is clearly dominated by horizontal and vertical displacement vectors. This holds also if border conflict constraints are applied. The inclusion of the displacement to the diagonal nodes will reduce the dominance, but comes at a higher computational cost, if the relationship between the nodes is not directly stored inside the quadnodes. Nevertheless, as the experiments of Section 8.2 have shown, real-time performance is still possible for data sets of considerable size.

The proposed solution differs from the solution by (Mackaness and Purves, 2001) as it does not apply displacement directly to a large number of map objects, but tries to accommodate additional point features to an already generalised point dataset that would have been omitted by the applied point reduction operator. Furthermore, the quadtree index enables to efficiently detect and localize conflicts and keep all operations confined to a local neighbourhood. The application of the *overlap minimisation function* to the candidate point feature for displacement would present an interesting combination of both approaches. A similar line of thought can be applied to the idea of the *least disturbing place* in the work by (Harrie et al., 2004), by additionally considering background constraints for point symbol placement (see also Section 6.7) and marrying both approaches.

Maintenance of spatial patterns The analysis of **if** and **how** the overall point distribution is maintained over a range of scales provides a quantitative insight to the distributional aspect of the proposed point generalisation algorithms. Section 8.8 quantitatively investigates the point distribution over a range of scales. The analysis is performed by analysing the development of density surfaces for each scale, compared to a reference scale, applying different methods of normalisation.

The analysis of the density differences and spatial variations of the point pattern, between the consecutive zoom levels and the selected reference map zoom level, shows that the spatial arrangement is mainly changed at locations with high point density. That is, where most of the cartographic conflicts are being addressed by the algorithm. It shows that the overall distribution is generally maintained. Low density regions may be overrepresented compared to the overall maxima, which can however be tackled by a different parameterisation of the algorithm, as shown in the experiments.

This observation is furthermore confirmed by the CHI expectation surfaces of the same map extent. The CHI expectation surface is built between the density of the generalised point pattern for the corresponding map zoom level and the expected density surface of the source point pattern, normalised to the Radical Law, and alternatively to the observed mean of the density of generalised point pattern.

For highly unevenly distributed point patterns, such as the one used in this evaluation, the challenge is manifested in the trade-off between the maintenance of the spatial coverage of the data vs. the distinctness of the spatial clusters.

Homogenisation The analysis of the homogeneity of the generalisation algorithm was presented in Section 8.9. Observations of the evolution of the nearest neighbour index NNI, with a more global explanatory character, show a relative increase of the NNI to a more regular pattern towards small scales.

On a local level, the comparisons of the nearest neighbour histograms and the G Function of the generalised and raw point patterns visualise the change between the two point patterns. If conflict constraints are fully resolved it shows no counts up to the symbol width; over longer distances, it gives hints that a regularity in the point pattern is introduced, particularly in midpoint aggregation. In terms of investigating the regularity of the pattern, the G function is most insightful and shows – similarly to the histogram – distinct breaks in the case of over-represented same distances (again, prominently in midpoint aggregation). Both the analysis of histograms and the G function allow the clear identification of regularities in the point pattern visible to the human eye, as is the case with the quadtree-based midpoint aggregation.

Cluster maintenance The final aspect of cartographic quality that was investigated, in Section sec:analysisClusterMaintenance, cartographic quality, is how well clusters of points are visually maintained over a range of scales. The analysis of the cluster maintenance first started off by identifying a density-based clustering method, DBSCAN, and its parameterisation based on a small participant survey. The participants were asked to rank from a set of maps those that best represent the clusters they would have drawn to highlight clusters on the map.

The epsilon distance chosen for the cluster analysis was based on the epsilon distance preferred by the test participants. The cluster analysis then compared quadtree-based point reduction (selection) and point displacement at one map zoom level, and for a representative operator (selection) the cluster maintenance over three consecutive map zoom levels. Additionally, the analysis considered for each algorithm the effect of *conflict constraints*, *grouping constraints* and the *symbol-to-node ratio* on the cluster maintenance.

A variation of the symbol-to-node ratio illustrates that a low ratio comparatively reduces the number of clusters. Comparing the number of clusters between selection and displacement shows that displacement retains more clusters than selection. The analysis of the quadtree-based selection shows that clusters tend to reduce in size and split, while the displacement algorithm maintains the shape of the clusters better. There, due to the working principle of displacement, clusters rather merge than split. This provides further evidence that the proposed displacement algorithm effectively refines the intermediate generalisation produced by point reduction algorithms (selection, simplification, aggregation).

The analysis of the cluster maintenance over two additional zoom levels also investigated the influence of different grouping constraints on the clustering. Generally, a reduction of map zoom results in a reduced number of clusters and an increased noise ratio, and the clusters tend to cover a larger geographic area. This holds for all applied conflict constraints.

The application of grouping constraints allows emphasising the most important clusters at a small scale, by an additional removal of points in low density areas. Thus, grouping constraints additionally reduce the number of points and increase the noise ratio, such that the clusters are more distinct. A second small user study showed that it is not the highest number of retained clusters that is being judged as the most preferred result, but a combination between noise ratio and cluster ratio. That is, the smaller the map zoom level, the higher the noise ratio has to be to keep the map clusters distinct and render with an acceptable result for the map user. This is particularly true if the map user gives preference to the maintenance of cluster centres higher over the overall spatial coverage of the point data.

10.3. Space-directed point generalisation

As a complement to the manipulation of point data, space-directed generalisation algorithms manipulate the map space. The conceptual classification of point generalisation algorithms introduced in Chapter 3 presents two transformation principles: focal transformation and the concept of a malleable space.

Focal transformations deform the underlying map space based on one or more predefined regions, often based on the metaphor of lenses. The malleable space concept adapts the map space locally to the existing map objects and is therefore a data-driven approach. The data-driven nature motivated the choice of the malleable space concept for the realisation of space-directed point generalisation in Chapter 7.

More precisely, the choice fell on the approach of Laplacian smoothing introduced by Edwardes (2007) and the diffusion cartogram algorithm by Gastner and Newman (2004), in order to test how existing space-directed algorithms can be integrated into the quadtree-based generalisation approach for on-the-fly point generalisation. Both methods rely on a well chosen, spatially weighted, initial parameterisation in order to avoid excessively deformed maps.

10.3.1. Contributions

For both methods, in order to enable space-directed point generalisation, the quadtree is being used as a 'binning' structure to guide the algorithm parameterisation. The quadtree index guides the weighting of the underlying weighted raster to control and limit the deformation of the map space.

Quadtree-based space-directed point generalisation uses the same definition for depth as in the object-directed solution. The weights – which essentially equate to local point density estimates – are therefore quadnode dependent. Hence, changing the zoom level changes the weights (or, the local densities), and therefore the degree of spatial deformation. At small scales the deformation is smoother, while for large scales the deformation is more detailed as there is a higher chance of weights (i.e. densities) changing.

The setting of the weights is solved by approaching the density distribution through the means of the quadtree. In the case of the cartogram the quadtree defines the bin size and the respective density. For Laplacian smoothing, that is less prone to undesired strong distortions, the quadtree approach allows to control the degree of deformation, by bin size and global weightings.

Space-directed generalisation deforms foreground and background elements together. That is, foreground point features and background map tiles. Background deformation, in our solution, is facilitated by the use of 3D texturing techniques applied to the deformation vectors.

Additionally, the quadtree is applied as a spatial index to combine object- and space-directed generalisation (cf. Chapter 9). The combination of object- and space-directed allows the retention of point data in areas of high point density and a reduction in the degree of generalisation in areas of low density. The pure application of space-directed operators, or their combination with object-directed operators, is facilitated through the same adaptable and modular approach, thanks to the unifying use of the quadtree index. This provides a great amount of flexibility and interaction possibilities for generalisation and exploration, data analysis and visualisation.

This thesis provides an initial implemented solution of a combined approach, and a set of further ideas on how to optimise the solution. The current implementation is, performance-wise, mainly restricted by the user-definable number of grid cells (Gastner and Newman, 2004; Edwardes, 2007), and real-time capability is reasonably achievable.

10.3.2. Insights and limitations

Section 3.3.2 presents a detailed discussion of the concept and comparison of the two methods used here for space-directed generalisation. Table 7.3 on page 105 provides a summary of the two methods and their individual differences in how they change the map space from a cartographic point of view.

Space-directed generalisation has in common with object-directed displacement that the number of points is not reduced. Thus, the range of scales at which both operators are applied is limited, in comparison to the point reduction operators.

Map deformation The deformation of the map space varies between cartogram and Laplacian smoothing, due to the different working principle of the two methods. Laplacian smoothing is less susceptible to strong distortions than the cartogram.

Furthermore, Laplacian smoothing allows controlling of the degree of deformation by setting a global weight and a smoothing factor. On the down side, depending on the spatial distribution, strong shrinking of cells is possible. While Laplacian smoothing shows no apparent directionality, depending on the point distribution the cartogram shows a directionality in the displacement given the method's density equalisation principle.

Finding a satisfactory parameterisation of space-directed algorithms is not quite straightforward, even with the help of a auxiliary data-driven index structure, and difficult to balance over a range of scales. Also, for the cartogram-based operator, while zooming out yields acceptable deformation characteristics, zooming into the dataset results in likely strong deformation.

Background scale The current implementation of the space-directed algorithms uses the current zoom level for the retrieval of the background tiles. For strong deformation, the local zoom level might be considerably increased, such that the zoom level of the loaded map tiles is too low (and the result looks pixellated).

One option to avoid this, is to assess the local scale by comparing the size and deformation of the resulting grid cells and to adapt the map tile resolution to the deformed map regions. There, the problem is finding an approach that provides a smooth change of scale, is not disturbing to the map reader's eyes and provides real-time performance. An *ad hoc* solution would be to select, for the map tiles, the scale with the strongest deformation or a scale midway, between the undeformed scale and the locally most deformed one. Yamamoto et al. (2009) present an interesting possible real-time solution, where the focus and context regions need to be assessed.

From the point of view of aesthetics and readability, it is advisable to select a background tile service that does not feature text elements. The current solution does not actively ensure the readability of rasterised text elements within the map tiles.

Combination of space- and object-directed generalisation The initial combination of object-directed quadtree-based generalisation operators (selection and centrality-based simplification) with space-directed quadtree-based generalisation shown in Chapter 7 provides a good first approach to demonstrate the strengths and likely further improvements of the method.

One point is to consider the local distortion of the map space, and adapt the depth of the object-based quadtree to the local map zoom level. Quadnodes with strong distortions or reduced area would then reduce the local content zoom level, while for increased node area the local content zoom level is increased.

Hence, the combined approach ideally allows to represent of more detail in regions of high point density, such as city centres, that would otherwise be generalised if the map space is not being deformed. This observation is in line with a comment made by Rhind (1983), who criticised about a population atlas that "‘the maps in ’People in Britain’ are the first reliable maps of unpopulated areas in Britain"’. Meaning that demographic variables could not be shown in sufficient details in densely populated areas, such as Greater London, while rural areas were over-represented in the choropleth maps used in that particular atlas.

Likewise, as in object-directed generalisation, border conflict resolution can be applied, to resolve overlaps of points at quadnode borders. Depending on the distortion of the individual quadnodes, second degree quadnode neighbours may be considered.

Edwardes (2007) also introduced a constraint formulation for Laplacian smoothing, to safeguard specific regions, such as linear features, from being deformed. Here, further investigation is required; for instance, the optimal range of scales of features that ’constrain’ the map space, and how much unconstrained space remains to deform the map space in a cartographically acceptable way.

User testing For both space-directed transformations, the perceptual and cognitive validity remains to be evaluated (Carpendale, 2001; Mountjoy, 2001; Lam et al., 2007; Schafer and Bowman, 2003; Pietriga et al., 2010). This is especially true, as compared to deformed, static maps, the dynamic nature of the deformation, user interaction, and heterogeneous data distribution are additional factors to consider in a web or mobile environment.

In order to carry out proper user testing the suggested improvements should be implemented, to avoid implementation effects affecting the user testing.

10.4. Methodology and workflow for real-time point data generalisation

The overall motivation for this work is driven by the ongoing transition in digital cartography and GIS from paper to screen to web to mobile maps. Hence, not only map requirements change, but also the context of usage, map interaction and performance. In the context of these changes map generalisation is a key process that enables a timely map usage. Thus, apart from providing generalisation algorithms, there is a need for overall methodology and workflows for point data generalisation.

10.4.1. Contributions

The proposed methodology of quadtree-based real-time point generalisation accounts for the changed requirements in mobile and web 2.0 applications and addresses the need for more interactivity, usability and modularity. Also, it provides a tighter link between cartographic generalisation and geographic visualisation.

This thesis offers a new definition of point generalisation in the context of web and mobile mapping. In chapter 3, it provided an extension of the typology for generalisation operators (McMaster and Shea, 1992; AGENT, 1999) and includes a space-directed transformation focus to the existing object-directed approaches. The classification bridges the existing views of how generalisation can be addressed: strictly object-directed or space-directed points of view. Hence, it allows the integration of algorithms into the generalisation process that deal with spatial transformation. Based on the typology, this work introduced an overall holistic methodology for mobile point generalisation.

The methodology delivers a generalisation process embedded in a problem-oriented workflow for point data generalisation. The workflow presents a flexible and modular solution to point data generalisation in real-time, and maps out how to integrate object- and space-directed generalisation operators into a common generalisation process, based on one influencing spatial index structure.

The modularity and applicability of the problem-oriented workflow for point data generalisation is presented on two case study examples and datasets (see Chapter 9). Furthermore, examples are given for simple label placements instead of point symbols. Chapter 9 also presents how special applications, such as spatial data mining or geostastical methods, can be integrated and combined with map generalisation, as well as the interplay between different data layers.

The thesis provides a novel approach, named *content zooming*, for information exploration in web and mobile maps, integrating several generalisation and zooming operations in order to better support mobile users in their information seeking and data exploration tasks. Content zooming serves as an example of additional interactivity given to user-map interaction, enabled by map generalisation itself. Content zooming takes a radically different perspective on map generalisation: it gives the user the choice to change the amount and granularity of foreground information presented, independently of the geometric map scale. It may not present the user with the cartographically best possible map solution, but in combination with real-time generalisation methods, it will provide the user with a more adaptable and exploitable visual representation of map content.

10.4.2. Insights and limitations

Overall conceptual classification of point generalisation algorithms Various criteria may be used to classify point generalisation algorithms. The classification introduced in Chapter 3 extends the typology of generalisation operators by (McMaster and Shea, 1992; AGENT, 1999; Edwardes et al., 2005) and follows the notion of Weibel and Dutton (1999) – in contrast to McMaster and Shea (1992) – where generalisation operators are independent of the data model applied by the generalisation algorithm. The distinction made between object- and space-directed algorithms is based on how the algorithms manipulate the map content, and feed into a hierarchical three-level classification.

The seemingly strict differentiation between object- and space-directed algorithms does not exclude a combination of these. For instance, object-directed displacement shares similarities, from an end map product point of view, with solutions of malleable space operators. This may be less the case with Laplacian smoothing, but more with the algorithms of variable-scale projection investigated by Edwardes (2007) for their application in map generalisation, as well as with the density-equalizing cartogram algorithm by Gastner and Newman (2004).

The focus of the classification is merely on point data generalisation. Hence, map generalisation of line, polygon or raster data is not covered in this classification (for an overview of existing classifications of generalisation operators, see Section 2.3). The point generalisation workflow then interacts with background vector and raster data through the formulation of background constraints (cf. Section 6.7).

POI data and point collections form the two ends of a spectrum characterising the types of foreground point data, where the position of the points is important (in POI) or more weight is given to the relative position among points (in point collections) (Burghardt et al., 2004b; Mannes, 2004). The distinction of point data types allows properly informing the selection of the generalisation operators and constraints.

Map representation modes In the generalisation workflow introduced in Chapter 3 the selection of the map representation mode precedes the generalisation process. The quadtree-based generalisation approach implements *proportional point symbol maps* and *symbol maps*, and does not provide the capability of changing the representation mode for specific scale ranges. To support alternative representation modes, especially for small scales, the system would have to be extended to include raster maps, density and contour maps.

Area class maps, such as choropleth, dasymetric and diagram maps (based on polygonial regions) probably cannot be supported in a straightforward way by the quadtree-based generalisation approach, as they require line simplification. Dot maps and point-oriented diagram maps are, however, may be possible. Chart maps would require a small change in the code of the aggregation and typification operators. It remains to be investigated whether it is understandable to a map reader which POI the diagrams apply to.

In addition to the inclusion of additional map representation modes to the quadtree-based generalisation approach, the scale ranges within which a representation mode is valid need to be defined and applied. These scale ranges, or according to Müller et al. (1995) *catastrophic levels*, require the definition of limits of applicability for different object classes (Cecconi and Galanda, 2002; Cecconi et al., 2002).

Generalisation process and content zooming The generalisation process (see Section 3.4.6) includes object-directed and space-directed generalisation algorithms. The combination allows chaining the operators together in various ways. Map interaction, generalisation constraints and spatial indices interplay with the generalisation process. Two interaction sequences (see Section 9.1) exemplify the flexibility of the approach. The selection of the quadtree as the unifying underlying spatial index simplifies the interplay between the operators and facilitates real-time performance.

While not all operators and constraints are fully implemented in the prototype application[1], Sections 9.2 and 9.3 present examples based on two selected case studies of applications of the generalisation workflow. The selection of the operators as well as the constraint formulation are not automatic and depend on the user. Free selection of operators and constraints probably work well in cases where a rather experienced user such as a data journalist or a researcher investigates a point dataset.

In a mobile environment with high occurrence of novice users pre-set generalisation strategies of a well defined data set, as presented in the chapter on content zooming, probably work better. Content zooming provides the user with the possibility of changing the granularity of the generalisation and the setting of constraints and parameters happens in the background. A good user interface design then would allow the user to quickly retrieve the requested map information.

Content zooming as a concept for the mobile and the 'desktop' environment provides a means of using map generalisation as a visual exploration tool. One of the limitations is the lack of user testing apart from the author's and her colleagues' experiences with the prototype system. Hence, further evaluation in addition to the quantitative analysis should include usability testing, such as a quantitative comparison of the different proposed strategies as well as a comparison to a map interface without the option of content zooming (with mere zooming, panning and selection operations).

10.5. Recalling the research questions

The main objective of this work was to integrate algorithms and hierarchical data structures for real-time generalisation and to investigate techniques of combining map generalisation and exploration. Based on the previous discussion of the results documented in the preceeding chapters, this section attempts to systematically answer the research questions formulated in Section 1.2.

[1] Background constraints and focal transformations are not implemented in the prototype application.

Real-time point generalisation algorithms: *What characteristics are essential for an efficient and flexible/modular real-time point generalisation algorithm?*

Existing approaches for real-time generalisation in the literature mainly rely either on pre-computation and hierarchical data structures or on fast generalisation algorithms that are based on simple heuristics. This dichotomy exhibits a trade-off between efficiency and flexibility that are based on simple heuristics. Both characteristics, however, are essential for on-line and real-time map generalisation. This work presents an approach that exploits hierarchical data structures but does not rely on pre-computation.

The approach taken employs the hierarchical properties of the quadtree, and uses this spatial index to inform local point generalisation algorithms. The data-driven nature of the quadtree provides the means of mapping the degree of map generalisation to the target map scale. Hence this approach does not rely on a further parameterisation algorithm that controls the amount of detail retained in generalisation.

The proposed generalisation algorithms benefit from further quadtree properties, such as topology and coverage, which enable neighbourhood queries and estimates on the point density and distribution. The generalisation algorithms implement all object-directed point generalisation operators of the classification for generalisation operators of Chapter 3. By using the same index structure and local implementation of the generalisation operators, the quadtree-based point generalisation approach is modular, as no change of the index structure is required if the operator is changed. The implementation features methods that ensure the maintenance of foreground-foreground and cartographic constraints. Methods to ensure foreground-background constraints were only provided as conceptual designs, as they require procedures that help support the understanding, on a semantic level, of the heterogeneous foreground and background data, in order to formulate constraints. Additionally, they would a good user interaction design to lead map users in the specification of constraints.

Generalisation evaluation: *What are the strengths and weaknesses of quadtree-based algorithms for real-time point generalisation and how do they perform?*

Chapter 5 introduces a diagnostic toolbox based on a set of measures to quantitatively assess the cartographic quality of point generalisation. Chapter 8 then presents the results of the quantitative cartographic analysis of the quadtree-based point generalisation algorithms, regarding several aspects of cartographic quality, including data reduction, conflict reduction, data enhancement, displacement effects, density variation, homogenisation and cluster maintenance.

The analysis shows that the approach generally fulfils the set cartographic constraints and executes generalisation in real-time. It reduces the amount of data according to the cartographic conflicts of a target scale, and maintains spatial patterns, the overall density distribution and large clusters. The analysis shows that the spatial arrangement is mainly changed at locations of high point density. This has the effect that regions of low point density are relatively over-represented; an aspect that can be addressed by setting the grouping constraints appropriatly to better retain cluster centres. This comes, however, at the cost of the overall spatial point coverage.

Space-directed point generalisation algorithms: *How can different types of point generalisation operators be integrated into a modular generalisation workflow?*

The typology of point generalisation operators proposed in Chapter 3 introduces a space-directed transformation focus. It includes a space-directed view to point map generalisation and allows the integration of spatial transformation algorithms into point generalisation. Chapter 7 presents solutions to space-directed point generalisation, while Chapter 9 illustrates how these space-directed operators can be integrated into a modular generalisation workflow.

Based on two algorithms that implement the malleable space concept, this thesis demonstrates how the diffusion cartogram algorithm (Gastner and Newman, 2004) and Laplacian smoothing (Edwardes, 2007) can be integrated with the quadtree-based approach, by exploiting the underlying grid structure, and inform the spatial deformation trough the quadtree. The quadtree is used as a hierarchical spatial index to derive scale-dependent density estimations to control the local deformation of the map space. The fact that the quadnodes that controls the deformation are the same as in the object-directed case, facilitates the combination of object- and space-directed generalisation operators. A combination of the principles allows inflating the map space in areas of high point density, and thus increase the level of detail that can be represented locally, and vice-versa.

Both algorithms deform map space including the background map, and perform, depending on a reasonable setting of the underlying grid, in real-time. Space-directed operators, if they are to maintain all points, are, similarly to the displacement operator, restricted in their application to a narrow range of scales. Also, the deformation of the map space does not *per se* resolve variations of conflict constraints. Finally, compared to object-directed generalisation, space-directed generalisation requires more effort in parameterisation and is not as straightforward to control. Additionally, the perceptual and cognitive validity and usability of this approach remains to be evaluated.

Map interaction and generalisation: *How can interaction in map generalisation be extended to enable dynamic online map information exploration?*

Interactive map exploration and visualisation are not the only processes to benefit from good map generalisation. Map generalisation can also be used as a means to facilitate visual exploration tasks, by applying the knowledge applied in map generalisation. This work introduced content zooming (Chapter 4) as a concept for mobile and desktop map interaction and a visual exploration concept. Content zooming takes a radically different perspective towards map generalisation: It enables the user to choose the amount and granularity of the foreground information presented, independent of the geometric scale. It provides the user with the option of changing the level of generalisation without 'losing' the current map context and extent. The different generalisation operators then provide different ways of changing and 'over-riding' the details, which as a consequence provides the user with a more adaptable and exploitable map content. User testing with a fully tested application is the next logical step to evaluate content zooming as an HMI method.

10.6. Recommendations

This work introduced and tested a complete set of algorithms for real-time point generalisation. All of the proposed algorithms are essentially based on the same quadtree data structure, and the presented workflows demonstrated how the various algorithms are applied and interact. However, since so many different individual results were presented, it may be difficult to distill the essence of this work. We therefore conclude this chapter by providing a list of recommendations that attempt to summarise the main findings regarding the usage of the proposed generalisation algorithms.

- Point reduction operators (selection, simplification, aggregation) are efficient in reducing clutter and complexity of point distributions, particularly when large scale changes are involved.
- If cartographic quality is important, then point reduction operators should always be followed by displacement, which efficiently removes overlap conflicts and congestions, and which allows to display more points.
- If maximum efficiency is key and if the cartographic quality is less important, point reduction algorithms should be used (without displacement), with quadnode border conflict resolution off, and with the caching option on.
- Value-based selection – depending on the selection criteria used – retains and visualises the local distribution of the attribute values.
- Midpoint aggregation – in combination with graduated symbols – is only useful in special applications, due to the homogeneity introduced in the point distribution.
- Quadtree-based point generalisation retains the coverage and overall arrangement of the spatial point distribution. The use of cluster constraints – especially for generalisation over a large range of scale – allows to emphasise clusters on the generalised maps.
- Both space-directed generalisation and object-directed displacement work well within a narrow band of scales. For an application over a wide band of scales, point reduction operators should precede these operators.
- The use of space-directed algorithms allows to spatially emphasize and inflate regions to provide more details in regions of high point density, while retaining the same overall spatial extent.
- Content zooming should be applied when one needs to adapt the amount of detail of the foreground map, while keeping the spatial context of the map background at the same map zoom level.

Chapter 11.

Conclusions

This thesis presented research on the generalisation of map content in web- and mobile mapping. The overall goal was to investigate how generalization operations for web and mobile devices can be made sufficiently fast while at the same time respecting principles of cartographic quality, so that the information portrayal can take place in real-time, in order that the scale and content of the map display can be flexibly and instantaneously adapted to the information requests and the context of a mobile user.

11.1. Main contributions

This work made contributions to the development of dynamic real-time generalisation of point data, and the integration of point generalisation with human computer interaction (HCI) techniques. Following is a summary of the main achievements of this thesis.

Methodology and workflow for real-time point generalisation This work identified and categorised methods for real-time generalisation and developed a comprehensive, conceptual methodology for real-time point generalisation, as well as a problem-oriented, modular workflow for point generalisation. It then introduced two main principles to approach point generalisation: the so-called object-directed operators, which manipulate map objects directly; and the space-directed operators, which manipulate and deform the map space in order to perform generalisation operations. The thesis proposed a suite of solutions for both object- and space-directed point generalisation operators.

Family of quadtree-based, real-time point generalisation algorithms The system implements both object- and space-directed point generalisation operators. The modularity and the real-time behaviour is reached by applying a hierarchical spatial index, the quadtree, to inform the generalisation algorithms. In addition, the use of the quadtree facilitates the implementation of generalisation constraints as well as methods for conflict resolution.

Based on the quadtree data structure, a large family of generalisation algorithms for all object-directed generalisation operators was developed, including selection, simplification, aggregation and displacement. In the case of space-directed generalisation, solutions to the concept of malleable space operators were presented. For both generalisation operations, Laplacian smoothing and the cartogram-based algorithms, the quadtree informs the space deformation based on a scale-dependent density estimation.

This is the first work that has presented a complete set of algorithms for all relevant operators of point data generalisation, operating in real-time, maintaining adequate cartographic quality. And it is the first work so far that made use of two paradigms of map generalisation, the object-directed and the space-directed approach. This has become possible because this work has systematically exploited beneficial properties of the quadtree data structure, and it has also been pioneering in this respect, since very few studies on map generalisation had made use of the quadtree before.

Content zooming as a map exploration tool This work furthermore contributes towards a more interactive use of map generalisation by introducing so-called 'content zooming' as a concept for visual exploration of content in web and mobile mapping. In combination with the quadtree-based point generalisation system, this solution to real-time generalisation not only illustrates the application of automated generalisation of point data, but also contributes to pushing interactive map generalisation towards a tool for visual map exploration.

11.2. Insights

Real-time map generalisation is key to web- and mobile mapping. The requirements for map generalisation differ depending on the output media and the context of usage. Furthermore, map generalisation also depends on the type and quality of the source map data that serves as an input to map generalisation.

Generalisation of point data has received limited attention in the literature so far, as the research in map generalisation has mainly concentrated on line and polygon generalisation. With the advent of web- and mobile mapping, however, and the possibility to mash-up diverse spatial data sources, point data – as the simplest but most abstract form of spatial data – have gained importance. Not only did the requirements of map generalisation change, but also the requirements of map interaction. Hence, apart from real-time behaviour, modularity to flexibly adapt the map content is a second main requirement in map generalisation for web- and mobile mapping.

Given these requirements – real-time behaviour, modularity and the point data type – a hierarchical approach is needed that enables fast heuristics and allows for a certain modularity. In the literature of map generalisation real-time generalisation is typically achieved by either relying on hierarchical and pre-computed data structures or fast and simple generalisation algorithms. While fast algorithms sacrifice cartographic quality for speed, pre-computed data structures lack flexibility as they are not capable of addressing changes in map specifications or content.

As was shown in this thesis, one solution to bridge the dichotomy between speed and flexibility is to use a hierarchical spatial index and map the hierarchical levels to the map scale. The point region quadtree proved to be an optimal candidate for a spatial index, providing all the properties needed to facilitate modular, real-time generalisation of point data. The regular subdivision and the data-driven nature of the quadtree facilitates the mapping between target zoom level and depth of the quadtree. The nodes of a quadtree at the target zoom level then provide the foundation to apply the generalisation operation on each subtree of these nodes. This allows maintaining the same spatial index principle to formulate different point generalisation operators. The solution provides modularity as it allows to implement all 'classic' generalisation operators on a local level, as well as the dynamic adaptation of these. This also holds for spatial transformations of the map with the space-directed generalisation paradigm.

The proposed approach to point generalisation is modular and extensible for further generalisation algorithms, space- as well as object-directed ones. The minimum requirements for point data generalisation are their coordinates. Additional attribute data and their 'scale' significantly extend the range to include further generalisation algorithms (e.g. value-based selection). If only the minimum requirements are met (i.e. if no point attributes are present), it is hard to properly formulate background constraints, as the semantics of the data, foreground and background data, are unknown *a priori*. However, background constraint rules can be formulated with well defined metadata or by user configuration, and with background data in the range of the target scale.

Content zooming and the modularity of quadtree-based point generalisation provide a lot of freedom to the map user to interact with the map content, which works well for a exploratory approach. In a mobile environment pre-set generalisation strategies (Chapters 4 and 9) are likely to provide a better user experience.

While the real-time and modular quadtree-based generalisation approach has its merits, it also exhibits certain limitations. For object-directed operators, the data-driven nature of the quadtree approach only applies generalisation where conflict constraints are violated; depending on the point pattern, homogenisation effects may occur at small scales. Grouping constraints reduce this effect, if the maintenance of cluster centres is preferred over maintaining the spatial coverage. For space-directed operators, finding satisfactory parameterisations over a range of scales is not straightforward.

In a further step, both object- and space-directed operators would benefit from an automated setting of the applicable range of scales – based on the point dataset, automated parametrisation, or a well-designed user interface for map generalisation.

11.3. Outlook

Methodology The increased availability of spatial data from various data sources, and a wide range of application contexts, call for new methods that automatically detect and interpret 'user generated' heterogeneous data (Sester et al., 2014). Generic approaches such as quadtree-based point generalisation presented in this thesis would benefit from the semantic enrichment of the data (Steiniger et al., 2006; Venkateswaran et al., 2014) to simplify the setting of the parameters and background constraints.

Also it is worth further investigating how to communicate possible map interaction modes – in terms of generalisation operators, content zooming and the setting of constraints – properly to the map user, through a highly responsive and intuitive user interface.

Extension of the quadtree-based approach The combination of further data types with the quadtree-based approach would be promising for future research, such as the adaptation of the line generalisation based on the Quaternary Triangular Mesh QTM (Dutton, 1999a,b), as both approaches share many similarities. Generally, locally operating line and polygon-based generalisation algorithms, with real-time or near real-time behaviour, are of interest. The combination of these with existing solutions for continuous generalisation (Cecconi, 2003; Sester and Brenner, 2004) and the investigation of interactions with web-based generalisation services (Neun, 2007) are also areas for future work.

Particular point data attributes Instead of extending the quadtree-based solution to other geometric data types, it might be interesting to develop further operators for particular types of point data and their attributes. An example of this could be explicitly including the temporal extent and then investigating how data points with a temporal scale should be generalised and how the process is likely to differ between static points from a sensor network (e.g. weather station), and dynamic points (e.g. animal observations). Another category of point data to investigate further are 'points' with associated texts, such as tags, descriptions or stories (Section 9.2).

User testing To better understand how content zooming and space-directed transformations are perceived and understood by the map user in the context of map generalisation, empirical user testing is required. Especially for both space-directed transformations, the perceptual and cognitive validity remains to be evaluated (Carpendale, 2001; Mountjoy, 2001; Lam et al., 2007; Schafer and Bowman, 2003; Pietriga et al., 2010). A main requirement, in addition to a proper study design, is a well-tested implementation with a reworked, well-designed user interface that is intuitive and highly responsive, in order to make sure the right 'variables' are measured.

Content zooming Several extensions of content zooming are conceivable. Extending content zooming for touch interfaces could be facilitated easily by integrating content zooming into a set of default touch gestures. Map zoom and content zoom might be implemented via spread and pinch gestures along the main axes of the screen. For instance, a horizontal spread gesture might be applied for geometrically zooming out on the map, while for zooming out on the content a vertical spread gesture might be used, with pinch gestures defined for the corresponding zooming-in commands. Content zooming could further be combined with focus+context methods, with which it shares some similarities. Additionally, a visual indicator which highlights whether at a certain LOD there is more detail available might be included, by either creating a responsive GUI element or by highlighting dense regions in a map if content zooming is activated. Further generalisation strategies, such as highlighting of co-occurrences using collocation rules of map features – such as restaurants and parking lots (Reichenbacher and De Sabbata, 2011) – are logical extensions. Finally, besides its use in personalised adaptation of mobile maps, as shown in the first case study in Section 4.3, content zooming can also be seen as a visual analytic tool, as illustrated in the second case study. In the "map use cube" of MacEachren and Kraak (1997) content zooming may be positioned as a tool that enables human-map interaction for revealing unknowns, driven by an individual user perspective.

Implementation The current implementation, which served as a test bench for all the presented algorithms and the cartographic analysis, maintains the whole dataset in the main memory. A more advanced option would be disk-based and possible include an external database to facilitate the handling of large data sources. Also, from an application point of view – especially for a web-based application – the creation of a JavaScript library would be a likely candidate for a light-weight point generalisation library for point data.

Appendix A.

Prototype implementation

This appendix describes the implementation of the **GenW2+** prototype. The prototype has been implemented to develop, test, analyse and test real-time generalisation algorithms for web- and mobile mapping. The application is a platform-independent stand-alone Java application. **GenW2+** is based on Java 1.6 and has been tested on Windows and OS X platforms.

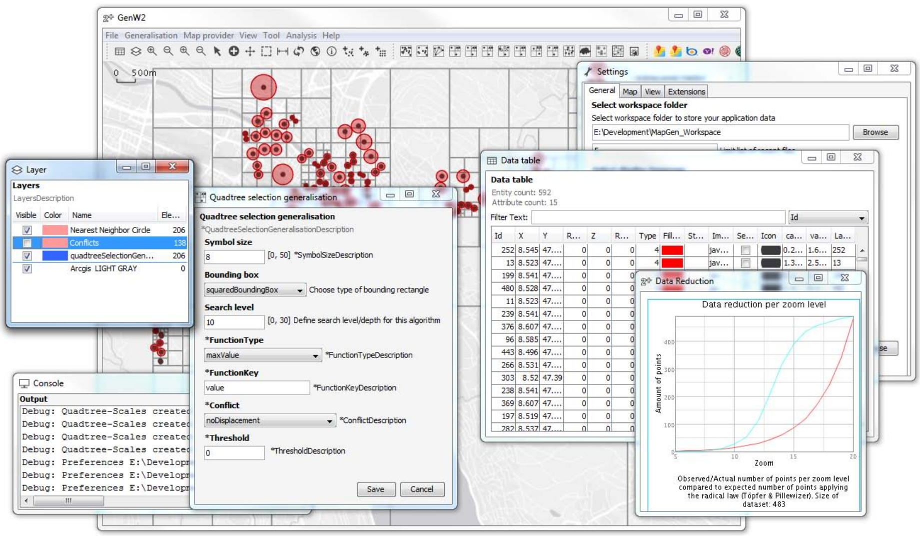


Figure A.1.: The GenW2+ development prototype

The following sections introduce first the class design and basis of the prototype which serves as a basis for the overall application, then an overview of the graphical user interface (GUI) is given, and finally the cartographic analysis part is introduced based on a set of illustrative examples.

A.1. The GenW2+ class design

The generalised UML class diagram in Figure A.2 provides an overview of the design of the prototype application, focusing on how the generalisation is separated from the overall system, not including the GUI and system relevant parts of the application.

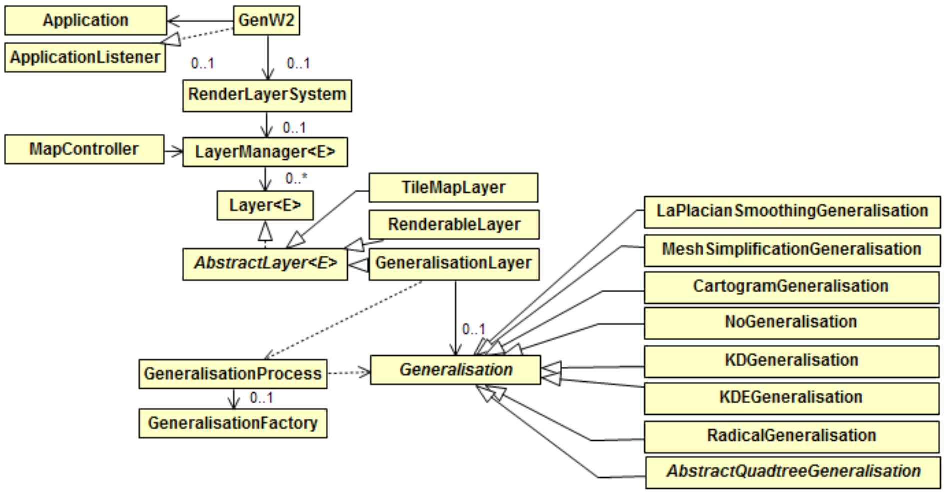


Figure A.2.: UML class design of the GenW2 prototype application. Abstract classes and interfaces are in italic.

GenW2+ implements an application interface, and among other systems instantiates a `RenderLayerSystem`. The `RenderLayerSystem` handles and renders the several map layers managed by the `LayerManager`. Among the different types of layers such as the `TileMapLayer` handling the background web map tile functionality or the simple `RenderableLayer` used for instance to render the results of the cartographic analysis, the `GeneralisationLayer` encapsulates the generalisation process. The different generalisation algorithms then implement the abstract `generalisation` class, allowing for a modular handling of the different generalisation algorithms.

A.2. Graphical User Interface

The platform supports a *map view* within the main screen and in a seperate window a *data view* of the currently loaded foreground data and its attributes (Figure A.3).

Map view The map view allows the user to explore the map in an interactive manner by zooming, panning and performing various actions. The map view provides the typical functionality and interaction of a web- or mobile mapping application. It features zooming and panning, essentially required to evaluated and test the various generalisation algorithms. Further interaction features have been implemented such as selection, object information, distance measurement and feature creation and editing tools.

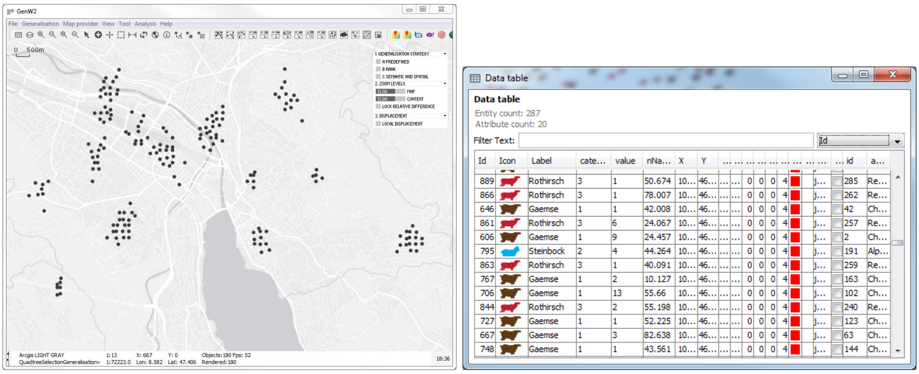


Figure A.3.: GenW2+ Main window with the map view (left) and the data view (right).

The implemented content zooming functionality and the content zooming workflow (see Chapter 4) is presented by a small GUI tool on the top right corner of the map view. In addition it is possible to manage the various map layers in a seperate window. The layers are typically the background layer with the map tiles, the foreground layer with the generalised data and further layers generated through cartographic analysis (see Section A.5).

Data view The data view presents the loaded data in a interactive tabular form. The data view features attribute-based search, editing of records, records selection and basic global statistics. Both views are linked, such that selected items are highlighted simultaneously in both views.

A.3. Main Window

The main window features a menu bar, toolbars, a status bar and the map window.

File menu Typically a file menu controls, file input and output, printing, and the application preferences. The prototype application reads simple comma-separated (csv) files storing point locations and associated attribute values, xml files (application based format) and Geo-RSS files (www.georss.org). The following listings in this section present examples of the different file formats for csv files (Listing A.2), XML encoded OSM point data (Listing A.1) and Geo-RSS (Listing A.3).

Listing A.1: Example OSM XML File containing one POI

```
<?xml version='1.0' encoding='UTF−8'?>
<osm version="0.6" generator="Osmosis 0.40.1">
  <bound box="47.05925,8.25954,47.79578,9.09429"
   origin="http://www.openstreetmap.org/api/0.6"/>
  <node id="123456" version="1" timestamp="1999−01−01T00:00:00Z" uid="999999"
```

```
    user="username" changeset="654321" lat="47.3637" lon="8.566">
      <tag k="name" v="Klusplatz"/>
      <tag k="amenity" v="bus_station"/>
      <tag k="operator" v="VBZ"/>
      ...
  </node>
</osm>
```

Listing A.2: CSV example of a geocoded flickr image

```
id, lat, lng, user, title, dateTaken, datePosted, tags
79..,46.774,9.842,XXX@N00,"Nice weather",11360...,1136...,"davos, skiing"
79..,46.799,9.826,XXX@N00,"Spengler Cup",11360...,1136...,"davos, spenglercup, hockey"
```

Generalisation menu The generalisation menu (see Section A.4) provides the different generalisation operators and algorithms, an individual parameter window for each algorithm and a debugging view option.

Map provider Different web map tile servers (WMTS) are supported, which provide background tile maps for the selected spatial reference. The implementation of the various WMTS servers has been extended based on the the *modest maps* library[1] for processing.

View This menu controls several view settings such as a fullscreen view and open further views such as the data view or the map layer view.

Tool The tool menu features a set of tools such as a console window for log data, a data provider view for GeoRSS data (Listing A.3), or plugins such as an R plugin for statistical analysis.

The data provider menu stores and manages URLs of GeoRSS providers to load their datasets.

Listing A.3: Example of an earthquake event stored in the GeoRSS Simple Atom Format

```
<feed xmlns="http://www.w3.org/2005/Atom" xmlns:georss="http://www.georss.org/georss">
  <title>USGS All Earthquakes, Past Week</title>
  <updated>2013−08−15T12:27:30Z</updated>
  <author>
    <name>U.S. Geological Survey</name>
    <uri>http://earthquake.usgs.gov/</uri>
  </author>
  <id>http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_week.atom</id>
  <link rel="self" href="http://earthquake.usgs.gov/earthquakes/feed/v1.0/summary/all_week.atom"/>
  <icon>http://earthquake.usgs.gov/favicon.ico</icon>
```

[1] Modest maps is a map library for processing (https://github.com/RandomEtc/modestmaps-processing), to display and interact with tile based maps, originating at Stamen design (stamen.com).

```
    <entry>
      <id>urn:earthquake—usgs—gov:ci:11350906</id>
      <title>M 1.4 — 10km N of Aguanga, California</title>
      <updated>2013—08—15T12:07:38.745Z</updated>
      <link rel="alternate" type="text/html" href="http://earthquake.usgs.gov/..."/>
      <summary type="html"> ... </summary>
      <georss:point>33.5372 —116.8455</georss:point>
      <georss:elev>—14900</georss:elev>
      <category label="Age" term="Past Hour"/>
      <category label="Magnitude" term="Magnitude 1"/>
    </entry>
  </feed>
```

Examples are the USGS earthquake hazards program (http://earthquake.usgs.gov) providing real-time earthquake news feeds[2] or news services geo-referenced by the *RSS to GeoRSS Converter* of geonames.org[3].

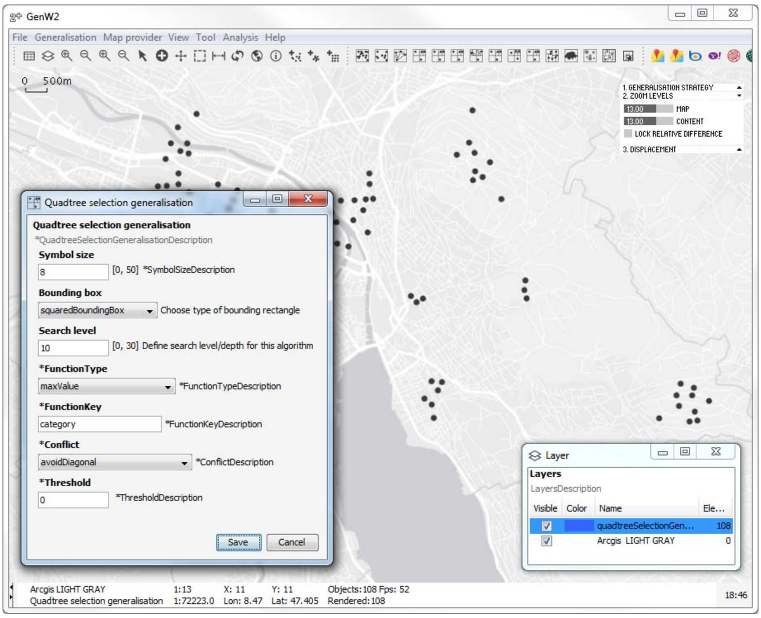


Figure A.4.: GenW2+ generalisation settings for the selection operator based on the quadtree.

The R plugin *RTool* is an interface for R to run R scripts to analyse the different generalisation algorithms (see Chapter 6). R (http://www.r-project.org) is a free software environment and language for statistical computing and graphics. The tool requires an

[2]USGS earthquake hazards GeoRSS: http://earthquake.usgs.gov/earthquakes/feed/v1.0/atom.php

[3]Geonames (http://www.geonames.org) is a geographical gazetteer database containing over 10 millions geographical names and features under a creative commons license.

installed R environment in order to run the selected R scripts.

Analysis The cartographic analysis menu provides a wide range of global and local measures and algorithms to analyse the generalisation algorithms quantitatively and qualitatively. The analysis results are provided in the form of statistical plots over a selected range of scales or as additional map layers in the map view.

A.4. Generalisation

As a development prototype for generalisation algorithms, special care has been given to the various generalisation algorithms (Figure A.4).

The implementation (see A.2) is designed in a modular way to encapsulate, if needed each algorithm. The user is given the possibility to either select among the different generalisation algorithms provided. Additonaly the prototype provides a debug view and for each of the algorithm the option to change the default parameterisation.

A.5. Analysis

The analysis and comparison of the various generalisation algorithms is provided by the cartographic analysis module. The analysis is presented in the form of numerical results, plots or as result layers in the map view.

Table A.1.: Cartographic analysis module and functions

Plot functions	
Information reduction	Data reduction (in comparison to Töpfer and Pillewizer (1966))
Feature displacement	Displacement count, Displacement magnitude, Displacement vectors
Spatial distribution	Standard distance, Spatial potential, Voronoi area, Delaunay area
Cartographic conflicts	Conflicts, Conflicts solved
Homogeneity	Nearest Neighbor Index NNI, Nearest Neighbor Distance
Entropy	Surkhov's, Neumann's, Bjørke's and Li & Huan's entropy measures
Density	KDE histogramm, KDE difference histogramm, KDE count, KDE difference count
Raster	Raster analysis histogram, Black pixel count
Performance	Performance all or one zoom level
Clusters	Optics reachability plot
Layer functions	
Analysis layer	Mean & median center, Standard ellipse, Conflict layer, Displacement vector layer, Nearest neighbor link, Nearest neighbor circle, Voronoi layer, Delaunay layer
KDE layer	KDE layer, KDE difference layer
Cluster layer	DBScan, Optics, KMeans
Raster layer	Raster layer, Raster counts layer

The plots provide the quantitative results for one scale or the evolution of a measure over a range of scales. The different plots can be exported as a PDF file and the possibility is provided to combine the results of different generalisation algorithms and parameterisations. An additional option provides results of the analysis in a tabular text format.

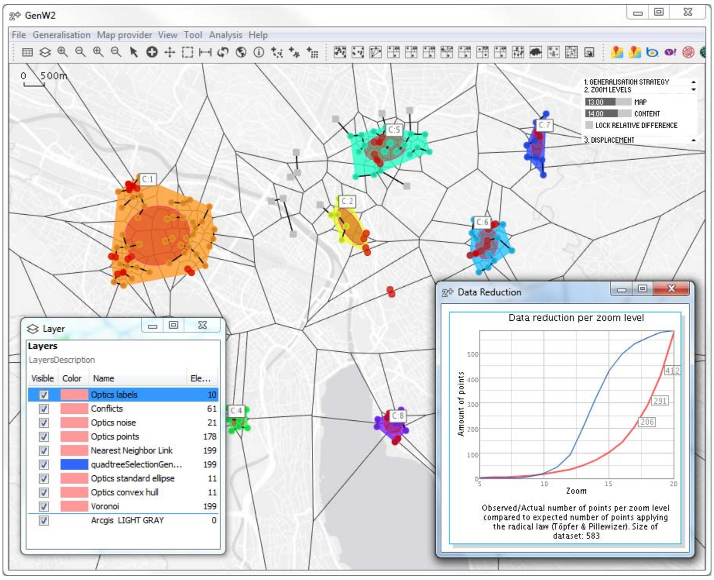


Figure A.5.: GenW2+ example of a generalisation analysis

An overview of the available analysis functions, plots and layers is given in Table A.1. These cartographic analysis module groups the measures based on the cartographic aspect measured. This are: *information reduction, feature displacement, spatial distribution, cartographic conflict detection and resolution, homogeneity, entropy, density, performance, clusters*. Chapter 6 gives an overview of the cartographic analysis module, while Chapter 8 demonstrates how these tools are used to systematically and comparatively assess the generalisation algorithms proposed in this thesis.

Additionally, the module features a plot reset function, various settings and the possibility to combine various plots in one plot result viewer.

Appendix B.

Datasets

The following paragraphs include information on the attributes and file formats of the datasets used in the experiments of this thesis.

The following datasets served as a basis for the cartographic analysis and the experiments performed in this work. This section organizes the data based on their main characteristics. The data formats used for the thematic point data layers were in most cases as tab-delimited text files or in XML data format.

Table B.1.: Overview of the datasets used in this thesis

	Data type	Area	# of points	Provider
POI				
OpenStreetmap[1]	POI	Zurich	>10000	OSM
Teleatlas	POI	Switzerland	54,912	Teleatlas
Swissnames	POI	Switzerland	155571	Swisstopo
Tourism Webcounts	'POI'	Switzerland	729	Venkateswaran (2010)
Swiss Activity dataset[1]	'POI'	Switzerland	576	Venkateswaran (2015)
PColl				
Animal observation[1]	PColl	ValFtur Switzerland	287	Swiss National Park
Lichen locality[1]	PColl	Switzerland	86000	SwissLichens WSL
Picture Location data	PColl	Switzerland	~2mio	Flickr API
Event-based PColl				
Earthquakes ECOS09[1]	PColl	Switzerland	6404	(Fäh et al., 2001)
Earthquakes	PColl	global	~20000/year	USGS
Twitter feeds	PColl	global	>8mio/day	Twitter API
RSS News feeds	PColl	global	~50/provider	Various news media
Artificial Datasets				
Uniform[1]		variable	variable	generated
Random[1]		variable	variable	generated
Clustered[1]		variable	variable	generated

[1] Datasets used in the experiments reported in this thesis.

B.1. Points of interest data (POI)

The points of interest data employed in this work originate from Openstreet Map.

OpenStreetMap POI OpenStreetMap (OSM) provides an extensive POI dataset. OSM is a collaborative comunity-based project, that creates and provides geographic data of the world. The data is licensed under the terms of the Creative Commons Attribution Share-Alike 2.0 license [1], which essentially allows to reuse and alter the data but subject to the condition to distribute the result under a similar license.

The data is available either as a weekly updated file *Planet.osm* containing all the data of OSM or as extracts for individual regions[2]. The two main data formats used are Protocolbuffer Binary Format PBF – intended as an alternative to XML – and human readable OSM XML (see Listing A.1).

For the experiments POI data for the bounding box of Switzerland was used. The data comes with a wide range of attributes and categories stored in key value pairs. For the use cases situated in the greater area of Zurich, attributes of type 'amenity' were extracted. The resulting datasets contains around 2700 points, with amenity attributes of categories such as restaurants, bars, cafés etc.

Tourism Webcounts The tourism webcounts datasets represents the online coverage of tourism webcounts for 729 populated places in Switzerland based on Swissnames. That is, for each populated place (small villages and cities) and search terms like 'tourism Zurich Switzerland' the dataset lists how many results where given by a web search engine. The dataset has been generated by Venkateswaran et al. (2014) to study the geographic and lingustic coverage of web resources in Switzerland. The search results were evaluated for the search terms in four different languages that is English, German, French and Italian. Tourism has been selected as a use case as it is specific to many places and vast information is available online in the form of news, reviews, blogs and the like.

Swiss activity dataset The activity dataset is a derived dataset that lists for each populated place in Switzerland activities associated to them. The activities are derived from 'tags', that is keywords, of geocoded pictures, taken in and near these places. The dataset therefore features for each place, a frequency list of all activity terms, the most frequent activities, how frequent they are, and further attributes. The generation and full description of the dataset is given in (Venkateswaran, 2015).

B.2. Point Collections

Point collections (PColl) encompass any point data that exist in large collections, such as count data or categorical observations collected at point locations, animal observations,

[1] See: http://www.openstreetmap.org/copyright and http://creativecommons.org/licenses/by-sa/2.0/

[2] A list of mirrors is available on the OSM Wiki pages (http://wiki.openstreetmap.org/wiki/Planet.osm) including a description of which region extracts are available, which formats and how frequently the data is being updated.

etc. (see also Section: 3.1.2). Animal observation data represent a PColl with point objects that potentially move. 'Find spots' of plants or the below presented lichens typically do not move. The following datasets where employed for the presented in the various experiments of this work.

Animal observations SNP The first dataset of this group features animal observation data in Val Ftur at the Swiss National Park (SNP). The dataset features 287 locations of cummulative animal observation – alpine ibex, chamois and red deer – with a total summed count of 1356 animal observation instances.

SwissLichens The second dataset originates from the Swiss Lichens database of the 'Nationales Daten- und Informationszentrum der Schweizer Flechten WSL'(Stofer et al., 2012). The dataset represents a point collection of lichen find-spots in Switzerland and maintains past and present population distribution of more than 500 different lichen species at over 86'000 locations within Switzerland. (A short introduction to lichen is given in Section 4.3.1; for an in depth information on lichens see Purvis 2000).

For each lichen find-spot the dataset features a large set of categories and attributes, as well as information on the precision, time and several indicator values such as the *red list status* (Scheidegger et al., 2002) or the *conservational priority* (BAFU, 2010) (see Table B.2 showing the available attributes).

Table B.2.: Extracted attributes and data from the SwissLichens database

Species & collection attribute	Species name Genus Ecotype Observation date
Locality	Coordinate (randomised $< 1km$ max) XY precision Habitat according to Delarze and Gonseth (2008) Substrate
Population characteristics	Size and vitality of the population
Ecology & conservation	Red list status (Scheidegger et al., 2002) NHV-Status (NHV) (Eidgenossenschaft, 2000) Conservational priority (BAFU, 2010) Ecological indicator value: Eutrophication (Wirth, 2010) Ecological indicator value: Humidity (Wirth, 2010) Ecological indicator value: Continentality (Wirth, 2010) Ecological indicator value: Light (Wirth, 2010) Ecological indicator value. Temperature (Wirth, 2010)

Earthquake catalogue of Switzerland ECOS The ECOS-09 (Fäh et al., 2001) is the earthquake catalogue of Switzerland and provides the combined and unified earthquake events of Switzerland and surrounding areas. It contains besides to other parameters the moment magnitudes (MW), coordinates of the epicentre of measured seismic events, the origin of the seismic event as being an earthquake and other induce seismic activity. The dataset covers all seismic events from 2001 to 2008 [3].

Earthquake events USGS The U.S. Geological Survey (USGS) provides a global catalogue[4] of locations and magnitudes of instrumentally registered earthquakes from 1900 to 2008. In addition a live GeoRSS earthquake event feed[5] provides information on the latest earthquakes registered and not older than 30 days (cf. Listing A.3).

Flickr image locations A third data source of PColl oriniates from the Flickr API, an online photo management and sharing service (www.flickr.com). The dataset consists of picture locations extracted from picture metadata for the area of Switzerland for the time span between 2005 and 2011.

The dataset contains around 2 million entries, with attribute information such as image id, latitude, longitude, user, image title, date taken and posted, and user attributed tags (keywords such as, sunset, zurich, lake. See listing A.2).

Live GeoRSS feeds A further point data source are live data news feeds in the form of geo-referenced RSS feeds of online news media containing typically one location per entry (cf. Listing A.3).

B.3. Artificial datasets

Artificially generated dataset were used as benchmark distributions of point data for an arbitrary area. Such point datasets were primarily used for debugging purposes, since they allowed to test the generalisation algorithms on very particular point distributions and spacial cases, such as densely clustered vs. completely uniform configurations. Examples of artificial datasets that were generated include *uniform*, *random* or *clustered* point distributions.

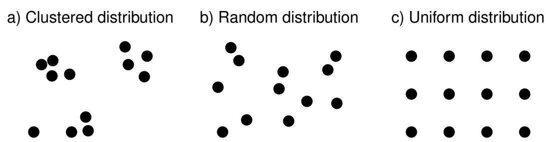


Figure B.1.: Examples of artificial point distributions

[3]ECOS Website: http://hitseddb.ethz.ch:8080/ecos09/index.html
[4]USGS centennial earthquake catalogue: http://earthquake.usgs.gov/research/data/centennial.php
[5]USGS earthquake hazards GeoRSS: http://earthquake.usgs.gov/earthquakes/feed/v1.0/atom.php

Appendix C.

Questionnaires

C.1. DBSCAN ε-neighbourhood questionnaire

This appendix shows the questionnaire used to investigate the settings of the ε-neighbourhood distance for the DBSCAN clustering algorithm. The results are presented in Table C.1, representing for each column the cluster maps (a-e) and for each row the possible rank given by the participants. That is, the first row represents to what degree the cluster maps have been given the first rank, and shows that in this participant group cluster map c with a ε-neighbourhood distance of 16, twice the symbol size, is preferred.

Table C.1.: Attributed epsilon rank (with $n = 7$)

Rank	a) 8	b) 12	c) 16	d) 20	e) 24
1.	0.14	0.14	0.43	0.29	0.00
2.	0.00	0.14	0.57	0.29	0.00
3.	0.29	0.43	0.00	0.14	0.14
4.	0.14	0.29	0.00	0.29	0.29
5.	0.43	0.00	0.00	0.00	0.57

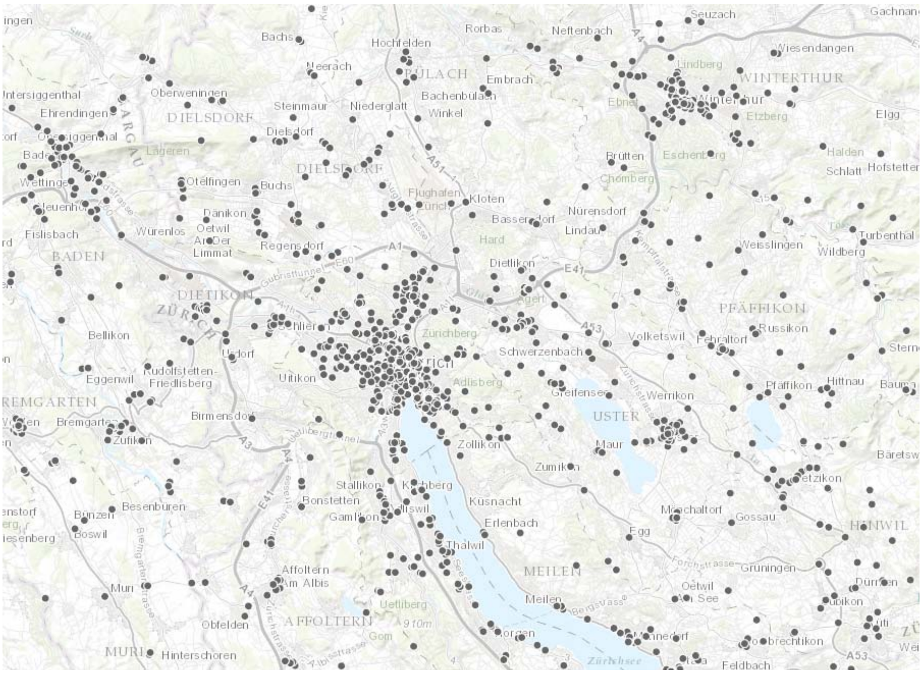
The expert users, however, had also the option to provide further comments and insights, as to why they had chosen a certain ranking, which are listed below, translated and in the original language:

1. D because according to my opinion spatially homogeneous distributions are represented well, E also but the 'Pfnüeselküste' (Western shore Lake Zurich) is quite strongly weighted.
2. For myself I prefer a further subdivision of groups that lie within cities, such that regions are recognisable within the cities (Winterthur). Zurich as a big cluster does not make sense.
3. This selection I did according to the given topic (restaurants). If it were hotels, I would have selected a different ranking.

In the original language:

1. *D da meiner Meinung nach räumlich homogene Verteilungen gut repräsentiert werden, E zwar auch aber 'Pfnüselküste' etwas gar stark gewichtet.*
2. *Ich für mich bevorzuge eine zusätzliche Unterteilung von Gruppen, die in Städten liegen, so dass auch innerhalb dieser Regionen erkennbar bleiben (Winterthur). Zürich als ein grosser Cluster ist nicht sinnvoll.*
3. *Questa scelta l'ho fatta secondo il tema che mi e stato dato. Se fosse stato un tema con hotel, avrei categorizzato diversamente.*

Clustering
Epsilon radius



Which of the following four cluster maps (A-E) (Epsilon settings) provide a good clustering in separating dense areas from the rest of the points?

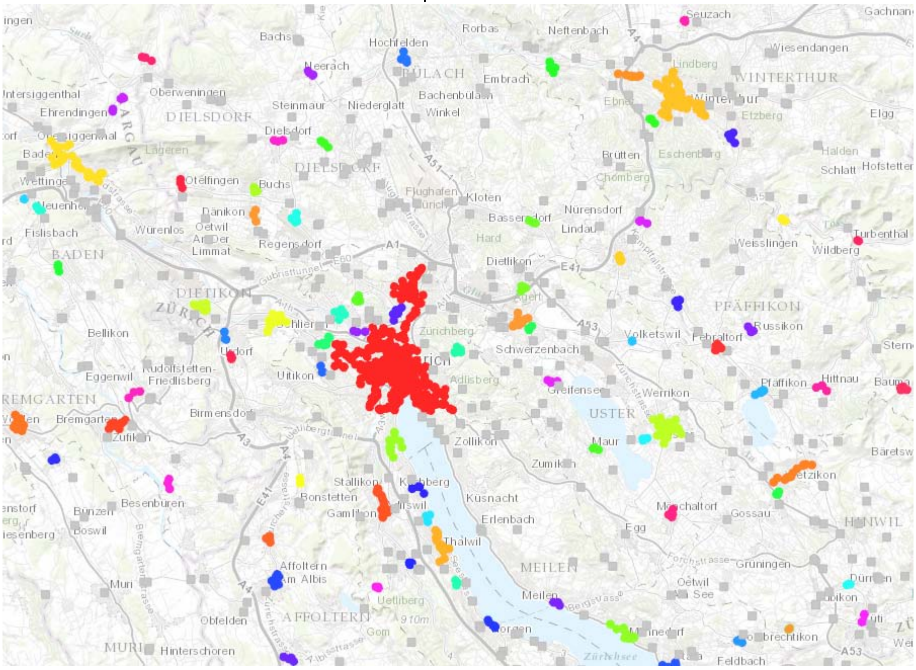
Please sort them in descending order of preference

1.
2.
3.
4.
5.

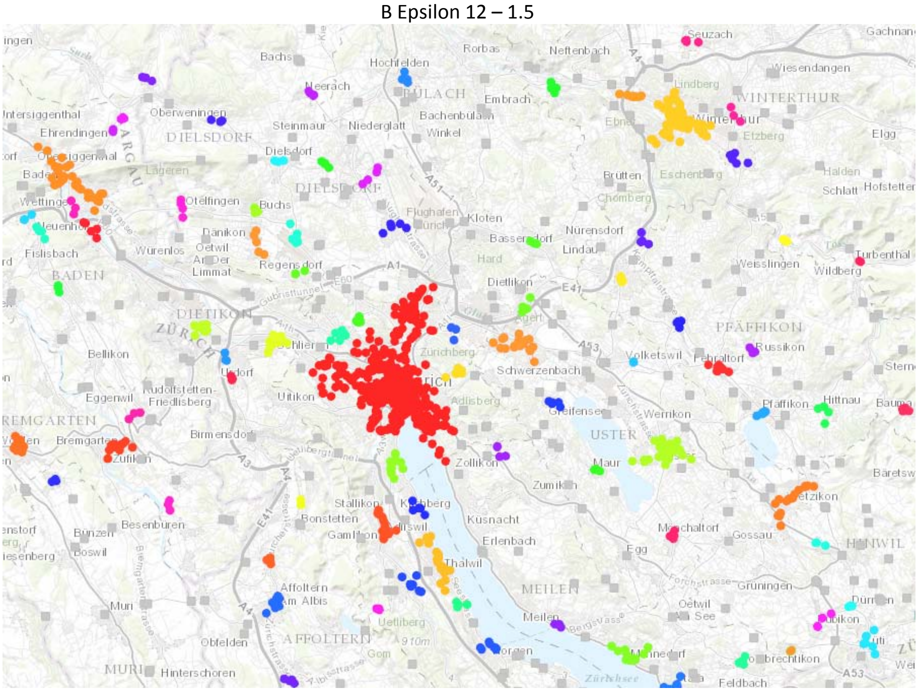
Please add some comments why you have chosen the selected cluster maps.

Please state where you see problems.

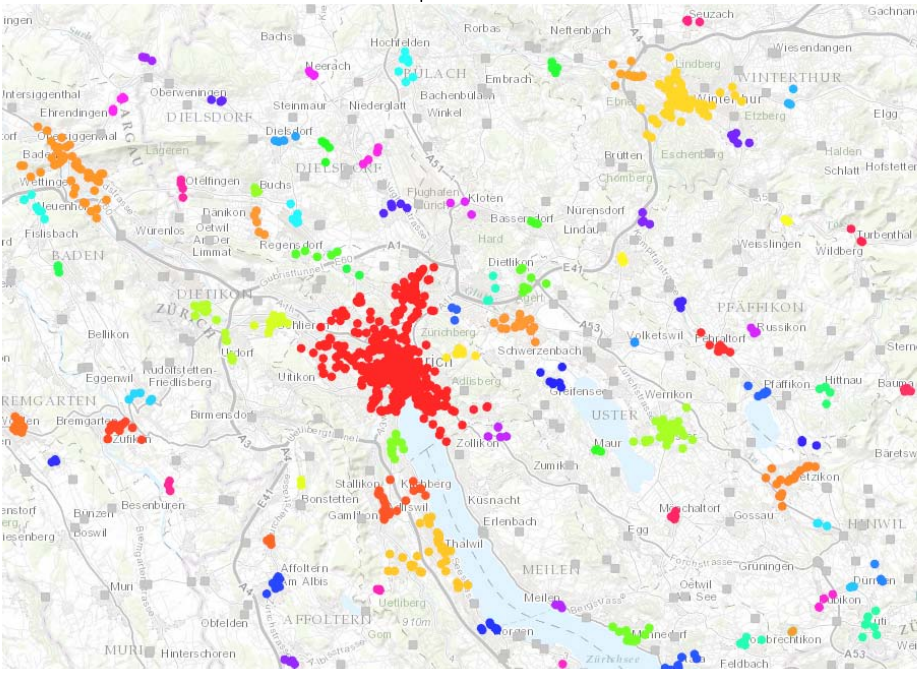
A Epsilon 8 - 1



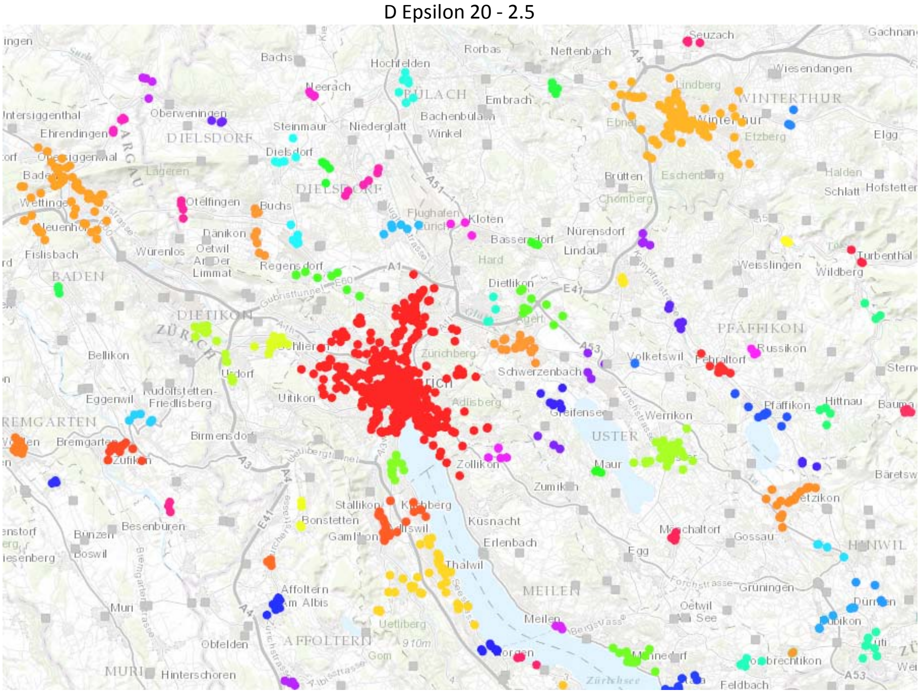
B Epsilon 12 – 1.5



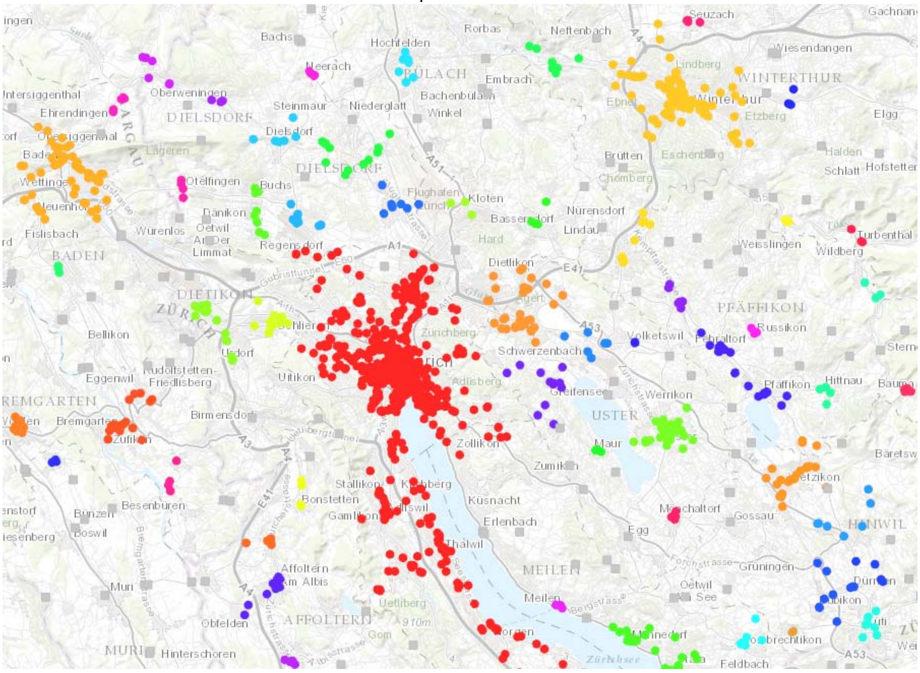
C Epsilon 16 - 2



D Epsilon 20 - 2.5



E Epsilon 24 - 3



C.2. Generalisation sequence questionnaire

This questionnaire was used to get an insight on the 'preferred' parameterisation of the quadtree-based generalisation algorithms over three consecutive zoom levels. Again the participants ($n = 7$) were asked to rank the generalised maps according to their preference. All three zoom levels show the same map extent (greater area of Zurich) using the quadtree-based selection algorithm (with conflict constraints *avoid*) with varying parameterisation.

For each task the parameterisation of the generalisation algorithm was slightly changed, such that the homogenisation at lower map zoom levels is less pronounced (see map zoom level 10 and 9 in the questionnaire).

The results of the three tasks are presented in Table C.2-C.6. Each column represents a generalisation solution that was presented to the participants, whereas the rows represent the normalised ranks given by each user.

Table C.2.: Task 1 Map zoom level 11 with grouping constraint p and d

Rank	a)	b) p = 1	c) d = 1
1.	0.43	0.57	0.00
2.	0.43	0.43	0.14
3.	0.14	0.00	0.86

Table C.3.: Task 2a Map zoom level 10 with grouping constraint $p = 1 - 4$

Rank	a)	b)	c)	d)
1.	0.29	0.57	0.14	0.00
2.	0.29	0.19	0.33	0.19
3.	0.29	0.19	0.48	0.05
4.	0.14	0.05	0.05	0.76

Table C.4.: Task 2b Map zoom level 10 with grouping constraint $d = 1 - 4$

Rank	a)	b)	c)	d)
1.	0.64	0.07	0.29	0.00
2.	0.21	0.64	0.00	0.14
3.	0.00	0.29	0.64	0.07
4.	0.14	0.00	0.07	0.79

Table C.5.: Task 3a Map zoom level 10 with grouping constraint $p = 1 - 4, 6, 8$

Rank	a)	b)	c)	d)	e)	f)
1.	0.07	0.21	0.05	0.33	0.33	0.00
2.	0.07	0.07	0.33	0.19	0.19	0.14
3.	0.14	0.00	0.23	0.23	0.23	0.18
4.	0.00	0.14	0.32	0.18	0.04	0.32
5.	0.14	0.50	0.04	0.04	0.18	0.11
6.	0.57	0.07	0.04	0.04	0.04	0.25

Table C.6.: Task 3b Map zoom level 10 with grouping constraint $d = 1 - 4, 6, 8$

Rank	a)	b)	c)	d)	e)	f)
1.	0.07	0.21	0.14	0.43	0.14	0.00
2.	0.21	0.21	0.29	0.29	0.00	0.00
3.	0.21	0.29	0.04	0.04	0.25	0.18
4.	0.07	0.14	0.46	0.18	0.11	0.04
5.	0.21	0.14	0.04	0.04	0.39	0.18
6.	0.21	0.00	0.04	0.04	0.11	0.61

The participants furthermore had the option to provide further comments and insights, as to why they had chosen a certain ranking, which are listed below (translated and in the original language):

1. 1: a,c unsure;2b:a,b,c similar,3a: e exactly right, slightly too few points but good, d,c,b,a, too many, 3b: f too few points
2. 2a: would rather like more points,3a: c-f too few points, in general: context important (geographic, personal) , most preferred as much information as possible
3. 2a too homogeneous, no cluster,d too few points, 2b: abc good (b=c) good because not overloaded and clusters are +- maintained, 3a: b too many points, a too homogeneous, ef in comparison to 3b f much better, 3b: Abc too many, ef too few
4. 3ab: don't like too thin or too opaque. General: at small scale I want overall guidance (skew ok) at large scale I want the information to be more complete

In the original language:

1. *1: a,c unsicher;2b:a,b,c ähnlich,3a: e genau richtig,etwas wenig Punkte aber gut, d,c,b,a, zu viel, 3b: f zu wenig punkte*
2. *2a: lieber mehr Punkte,3a: c-f zu wenig Punkte, allgemein: kontext wichtig (geographisch, persönlich) am liebsten soviel infos wie möglich*
3. *2a: a zu homogen, keine Cluster,d zu wenig, 2b: abc gut (b=c) gut da nicht überladen und cluster +- erhalten, 3a: b zu viel, a zu homogen, ef im Vergleich zu 3b f viel besser, 3b: abc zu viel, ef zu wenig.*

Clustering
Generalisation Sequence

In the following generalisation sequences from map zoom level 11 to 9 different parametrisation settings have been chosen.

Please select for each zoom level which solution you liked most in descending order.

Map zoom level 11 (a-c)
Task 1:

1.
2.
3.

Map zoom level 10

Task 2A (a-d)
1
2
3
4
Comments:

Task 2B(a-d)
1
2
3
4
Comments:

Map zoom level 9

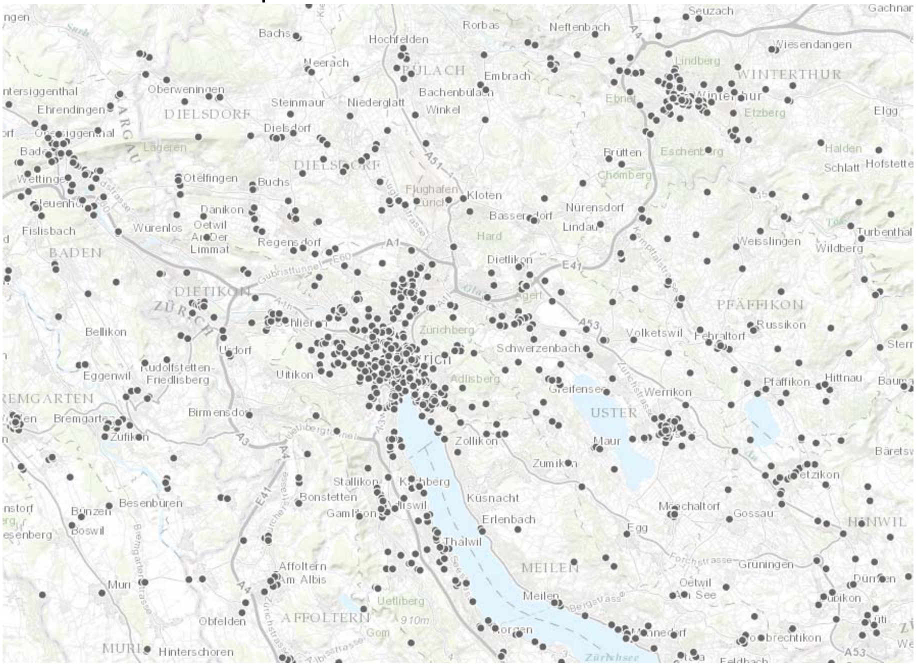
Task 3A(a-f)
1
2
3
4
5
6
Comments:

Task 3B(a-f)
1
2
3
4
5
6
Comments:

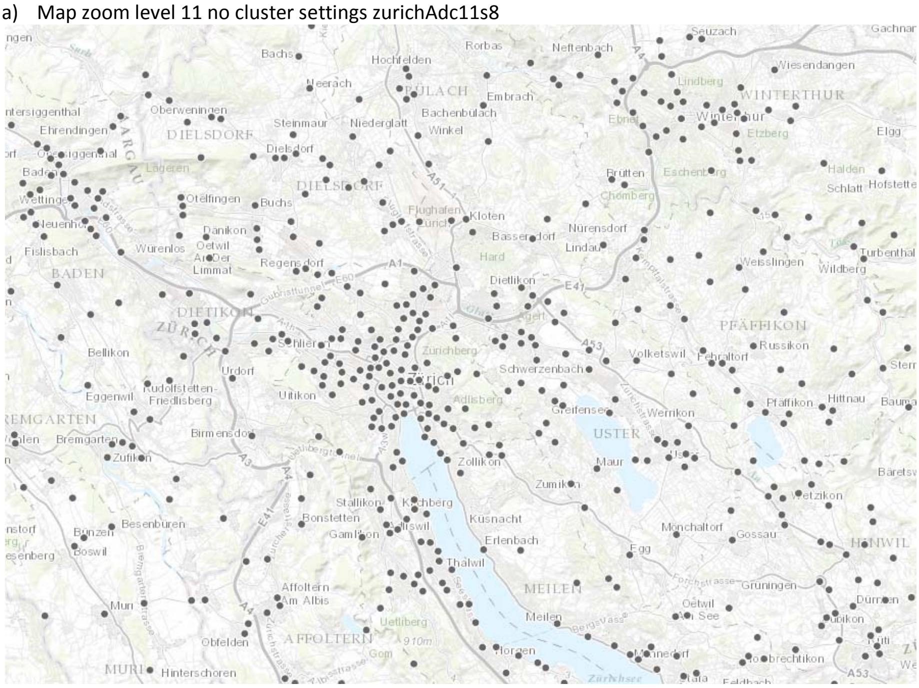
Please add some comments why you have chosen the selected maps.

Please state where you see problems.

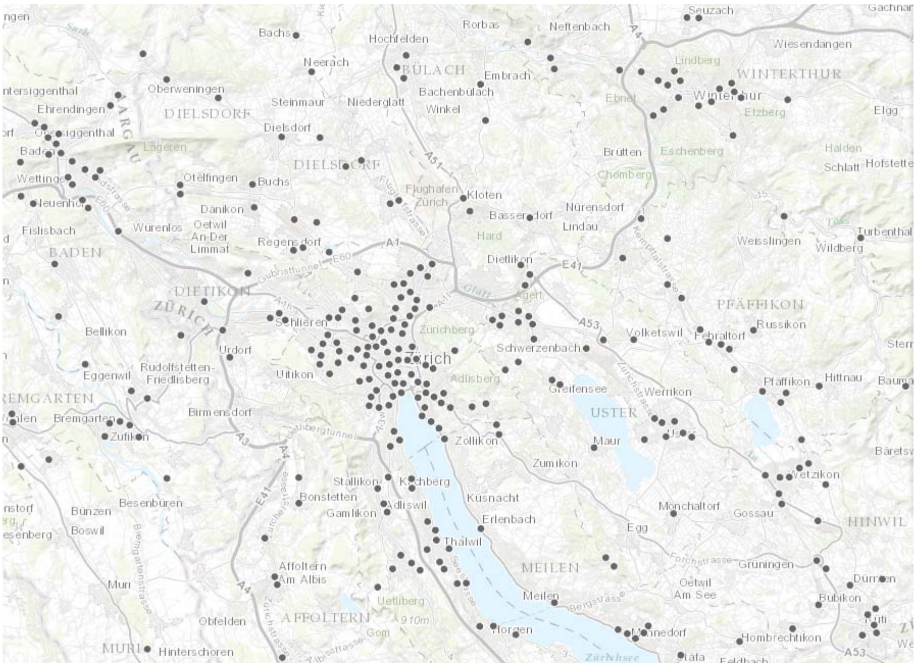
TASK 1: Raw data set for comparison



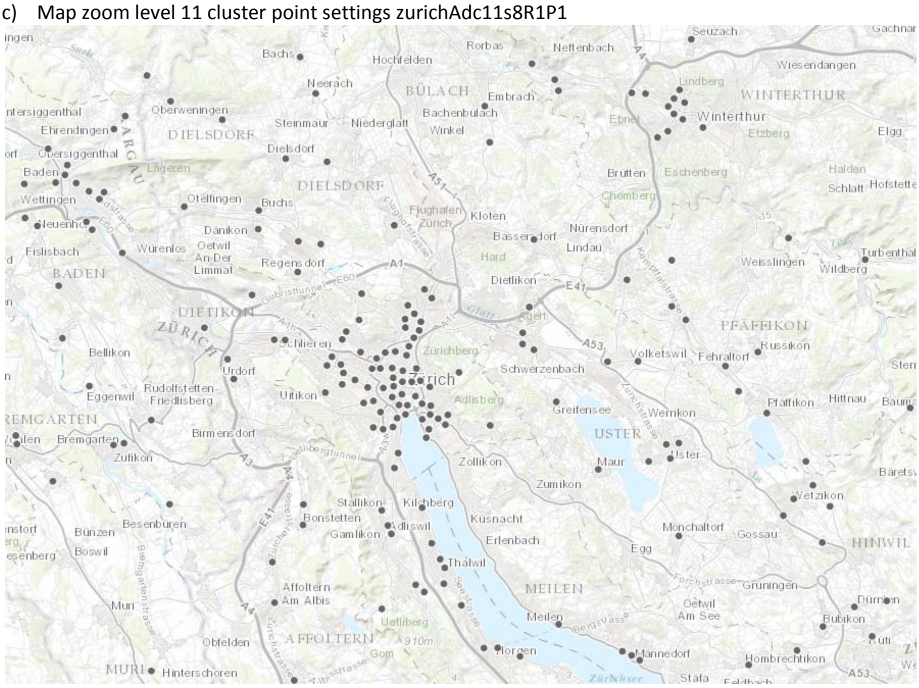
a) Map zoom level 11 no cluster settings zurichAdc11s8



b) Map zoom level 11 cluster point settings zurichAdc11s8R0P1

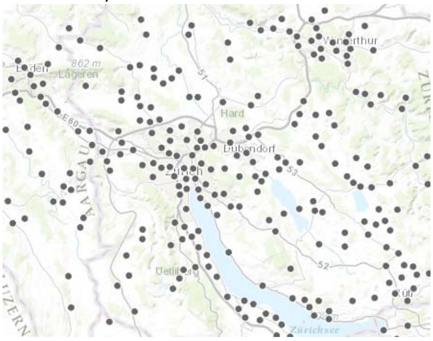


c) Map zoom level 11 cluster point settings zurichAdc11s8R1P1

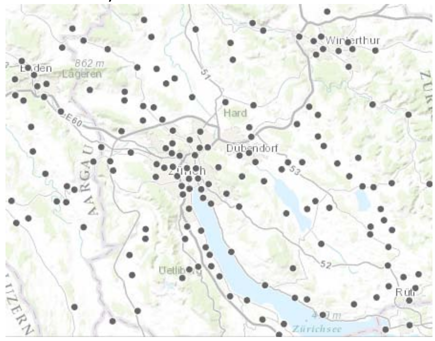


Task 2A: Map zoom level 10 cluster points parameter

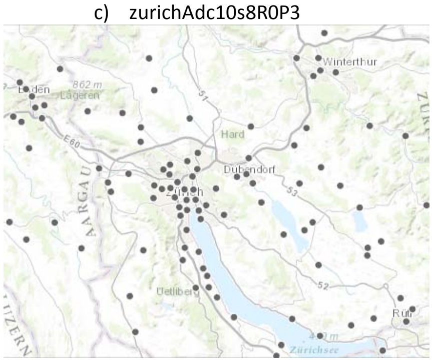
a) zurichAdc10s8R0P1



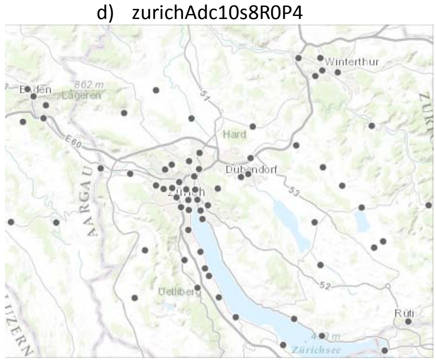
b) zurichAdc10s8R0P2



c) zurichAdc10s8R0P3

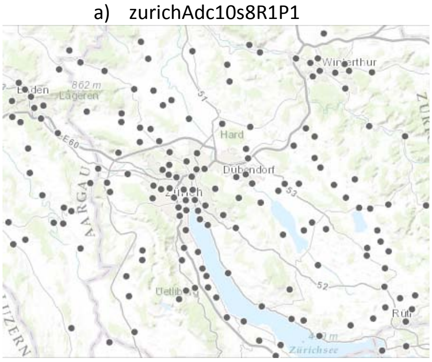


d) zurichAdc10s8R0P4

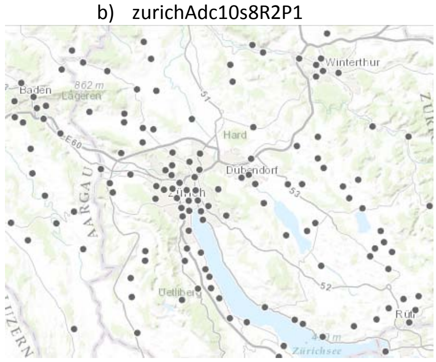


Task 2B: Map zoom level 10 cluster depth parameter

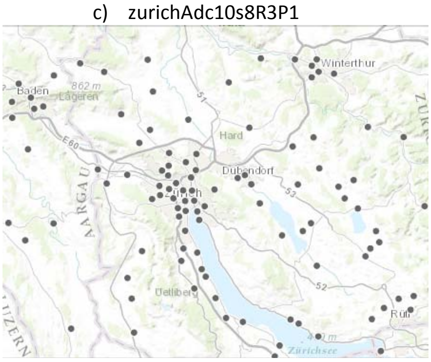
a) zurichAdc10s8R1P1



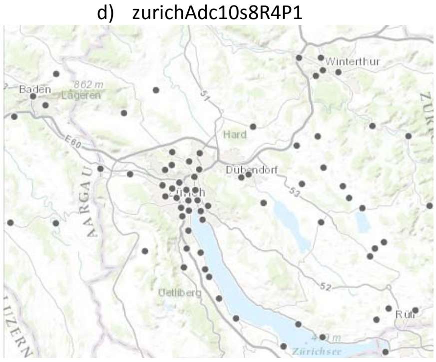
b) zurichAdc10s8R2P1



c) zurichAdc10s8R3P1

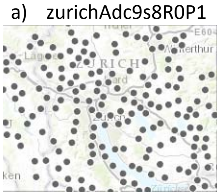


d) zurichAdc10s8R4P1



Task 3A: Map zoom level 9 cluster points parameter

a) zurichAdc9s8R0P1



b) zurichAdc9s8R0P2

c) zurichAdc9s8R0P3

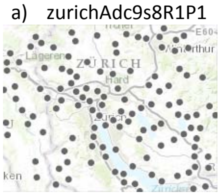
d) zurichAdc9s8R0P4

e) zurichAdc9s8R0P6

f) zurichAdc9s8R0P8

Task 3B: Map zoom level 9 cluster depth parameter

a) zurichAdc9s8R1P1



b) zurichAdc9s8R2P1

c) zurichAdc9s8R3P1

d) zurichAdc9s8R4P1

e) zurichAdc9s8R6P1

f) zurichAdc9s8R8P1

Complete publication list

The research carried out in the course of this thesis resulted in a set of additional publications that have not explicitly been introduced in this monograph. These additional publications are marked with a star *.

Journal Papers

Bereuter, P. and Weibel, R. (2013). Real-time generalization of point data in mobile and web mapping using quadtrees. *Cartography and Geographic Information Science*, 40(4):271–281.

Bereuter, P., Weibel, R., and Burghardt, D., (2013). Content zooming and information exploration for mobile maps. *International Journal of Geomatics and Spatial Analysis*, Revue internationale de géomatique, 23(3-4), 295–321.

Conference and Workshop Papers with Full Paper Review

Bereuter, P. and Weibel, R. (2010). Generalisation of point data for mobile devices: A problem-oriented approach. In S. Mustiere and W.A. Mackaness, editors, *13th Workshop on Progress in Generalisation and Multiple Representation*, *ICA Commission on Generalisation and Multiple Representation*, pages 1–8. Zurich, Switzerland.

Bereuter, P., Weibel, R., and Burghardt, D. (2012). Content zooming and exploration for mobile maps. In: J. Gensel, D.J. And, and D. Vandenbroucke, eds. *Proceedings of the AGILE 2012 International Conference onf Geographic Information Science*. Avignon, 74–80. (short paper)

Bereuter, P. and Weibel, R. (2012). Assessing Real-Time Generalisation of Point Data. In *16th Workshop on Progress in Generalisation and Multiple Representation map generalization.*, Dresden, Germany, 1–9.

Conference Papers with Abstract Review

*Bereuter, P., Venkateswaran, R., and Weibel, R. (2009). The use of filters for adaptive mobile mapping scenarios. In Tomko and K. Richter, editors, *Proc. AGILE 2009 Workshop on Adaptation in Spatial Communication*, pages 39–44, Hannover, Germany.

Venkateswaran, R. and Bereuter, P. (2009). User Adaptive Trip Planner. In: Ordnance Survey Geospatial Mashup Challenge – GISRUK 2009. Durham, UK, 1–5. (short paper)

Bereuter, P. and Weibel, R. (2011). A Diagnostic Toolbox for Assessing Point Data Generalisation Algorithms. In *Proceedings of the 25th International Cartographic Conference.*, Paris, France, 1–10.

Bereuter, P. and Weibel, R. (2012). Algorithms for on-the-fly generalization of point data using quadtrees. In *Proceedings of the 17th AutoCarto conference.*, Columbus, OH, 1–16.

Book chapter

Stanislawski, L. V, Buttenfield, B., Bereuter, P., and Brewer, C., 2014. Generalisation Operators. In: D. Burghardt, C. Duchêne, and W. Mackaness, eds. *Abstracting Geographic Information in a Data Rich World: Methodologies and Applications of Map Generalisation*, Springer International Publishing, 157–195.

Poster

*Venkateswaran, R., Bereuter, P., Burghardt, D., and Weibel, R. (2009). The GenW2 Framework: An Architecture for Mobile GIS and Mapping Scenarios. In: AGILE 2009. Hannover, Germany: Leibnitz Universität Hannover, 1–6.

Bibliography

AGENT (1998). Constraint Analysis. Deliverable A2. Technical report, Department of Geography, University of Zurich, Zurich.

AGENT (1999). Selection of Basic Algorithms. Deliverable D2. Technical report, University of Zurich, Zurich.

Anders, K. and Bobrich, J. (2004). MRDB approach for automatic incremental update. In *ICA Workshop on Generalisation and Multiple Representation*, page 8, Leicester, UK.

Anders, K. H. (2003). A hierarchical graph-clustering approach to find groups of objects. In *Proceedings 5th Workshop on Progress in Automated Map Generalization*, pages 1–8. Citeseer.

Ankerst, M., Breunig, M. M., Kriegel, H.-P., and Sander, J. (1999). OPTICS - Ordering Points to Identify the Clustering Structure. In *Proceedings of ACM SIGMOD International Conference on Management of Data (SIGMOD'99)*, volume 28, pages 49–60, Philadelphia:. ACM.

Bader, M. (2001). *Energy Minimization Methods for Feature Displacement in Map Generalization*. Phd thesis, University of Zurich.

Bader, M., Barrault, M., and Weibel, R. (2005). Building displacement over a ductile truss. *International Journal of Geographical Information Science*, 19(8-9):915–936.

BAFU (2010). Liste der National Prioritären Arten, Arten mit nationaler Priorität für die Erhaltung und Förderung. *Umwelt-Vollzug Nr. 1103*, page 132.

Bak, P., Schaefer, M., Stoffel, A., Keim, D. A., and Omer, I. (2010). Density Equalizing Distortion of Large Geographic Point Sets. In Purves, R., Clough, P., and Jones, C., editors, *Cartography and Geographic Information Science*, volume 36, pages 237–250. ACM.

Bard, S. (2004a). *Méthode d'évaluation de la qualité de données géographiques généralisées - Application aux données urbaines*. Phd thesis, Université Paris 6.

Bard, S. (2004b). Quality Assessment of Cartographic Generalisation. *Transactions in GIS*, 8(1):63–81.

Barrault, M., Regnauld, N., Duchène, C., Haire, K., Baeijs, C., Demazeau, Y., Hardy, P., Mackaness, W. A., Ruas, A., and Weibel, R. (2001). Integrating multi-agent, object-oriented and algorithmic techniques for improved automated map generalization. In *Proceedings 20th International Cartographic Conference*, number 1, pages 2110—-2116, Beijing, China.

Beard, M. K. (1991). Constraints on rule formation. In Buttenfield, B. and McMaster, R., editors, *Map generalization: making rules for knowledge representation*, chapter 7, pages 121–135. Longman Scientific & Technical, Essex, UK.

Berendt, B., Barkowsky, T., Freksa, C., and Kelter, S. (1998). Spatial representation with aspect maps. *Spatial cognition - An interdisciplinary approach to representing and processing spatial knowledge*.

Bereuter, P. and Weibel, R. (2010). Generalisation of point data for mobile devices: A problem-oriented approach. In Mustiere, S. and Mackaness, W., editors, *13th Workshop on Progress in Generalisation and Multiple Representation*, pages 1–8, Zurich. ICA Commission on Generalisation and Multiple Representation.

Bereuter, P. and Weibel, R. (2011). A Diagnostic Toolbox for Assessing Point Data Generalisation Algorithms. In *Proceedings of the 25th International Cartographic Conference*, pages 1–10, Paris, France.

Bereuter, P. and Weibel, R. (2013). Real-time generalization of point data in mobile and web mapping using quadtrees. *Cartography and Geographic Information Science*, 40(4):1–11.

Bertolotto, M. and Egenhofer, M. J. (2001). Progressive transmission of vector map data over the world wide web. *GeoInformatica*, 5(4):345–373.

Bivand, R., Pebesma, E., and Gomez-Rubio, V. (2008). *Applied spatial data analysis with R*, volume 65 of *Use R!* Springer Science+Business Media, LLC. 2008.

Bj, J. (1996). Framework for entropy-based map evaluation. *Cartography and Geographic Information Science*, 23(2):78–95.

Bobzien, M. and Burghardt, D. (2008). Multi-representation databases with explicitly modeled horizontal, vertical, and update relations. *Cartography and* . . . .

Boots, B. and Okabe, A. (2007). Local statistical spatial analysis: Inventory and prospect. *International Journal of Geographical Information Science*, 21(4):355–375.

Brassel, K. E. and Weibel, R. (1988). A review and conceptual framework of automated map generalization. *International journal of geographical information systems*, 2(3):229–244.

Brewer, C. A. and Buttenfield, B. P. (2007). Framing Guidelines for Multi-Scale Map Design Using Databases at Multiple Resolutions. *Cartography and Geographic Information Science*, 34(1):3–15.

Burghardt, D. (2005). Controlled line smoothing by snakes. *GeoInformatica*, 9(3):237–252.

Burghardt, D. and Cecconi, A. (2007). Mesh simplification for building typification. *International Journal of Geographical Information Science*, 21(3):283–298.

Burghardt, D., Edwardes, A., and Mannes, J. (2004a). An architecture for automatic generalisation of mobile maps. *2nd Symposion on Location based service*.

Burghardt, D. and Meier, S. (1997). Cartographic displacement using the snakes concept. In Förstner, W. and Plümer, L., editors, *Semantic Modeling for the Acquisition of Topografic Information from Images and Maps*, chapter 4, pages 59 – 71. Birkhäuser Verlag, Basel, Switzerland.

Burghardt, D., Neun, M., and Weibel, R. (2005). Generalization Services on the Web - Classification and an Initial Prototype Implementation. *Cartography and Geographic Information Science*, 32(4):257–268.

Burghardt, D., Purves, R., and Edwardes, A. (2004b). Techniques for on the-fly generalisation of thematic point data using hierarchical data structures. In *Proceedings of the GIS Research UK 12th Annual Conference*, Norwich, UK.

Burghardt, D., Schmid, S., and Stoter, J. (2007). Investigations on cartographic constraint formalisation. In *Workshop of the ICA Commission on Generalisation and Multiple Representation*, Moscow.

Butler, D. (2006). Mashups mix data into global service. *Nature*, 439(7072):6–7.

Buttenfield, B. P. and Wolf, E. B. (2007). 'The Road and the River Should Cross at the Bridge' Problem: Establishing Internal and Relative Topology in an MRDB. In *10th ICA Workshop on Generalization and Multiple Representation*, number August, pages 1–12, Moscow.

Carpendale, M. (2001). Examining comprehension issues in elastic presentation space. *Information Design Journal*, 10(1):58–68.

Carpendale, M., Cowperthwaite, D., and Fracchia, F. (1997). Making distortions comprehensible. In *Proceedings. 1997 IEEE Symposium on Visual Languages (Cat. No.97TB100180)*, pages 36–45. IEEE Comput. Soc.

Cecconi, A. (2003). *Integration of cartographic generalization and multi-scale databases for enhanced web mapping*. Phd thesis, University of Zurich.

Cecconi, A. and Galanda, M. (2002). Adaptive zooming in Web cartography. In *Computer Graphics*

Forum, volume 21, pages 787–799. Amsterdam, Holland.

Cecconi, A., Weibel, R., and Barrault, M. (2002). Improving automated generalization for on-demand web mapping by multiscale databases. In Richardson, D. E. and van Oosterom, P., editors, *Advances in Spatial Data Handling*, number Peterson 1999 in Advances in Spatial Data Handling, chapter 38, pages 515–531. Springer, Berlin.

Cheung, Y. K., Li, Z., and Chen, W. (2009). Integration of Cognition-based Content Zooming and Progressive Visualization for Mobile-based Navigation. *The Cartographic Journal*, 46(3):268–272.

Christensen, J., Marks, J., and Shieber, S. (1995). An empirical study of algorithms for point-feature label placement. *ACM Transactions on Graphics*, 14(3):203–232.

Clark, P. J. and Evans, F. C. (1954). Distance to Nearest Neighbor as a Measure of Spatial Relationships in Populations. *Ecology*, 35(4):445.

Cockburn, A., Karlson, A., and Bederson, B. B. (2006). A Review of Focus and Context Interfaces. Technical Report 1, Department of Computer Science, University of Maryland, Maryland, MD, USA.

Comer, D. (1979). The Ubiquitous B-Tree. *ACM Computing Surveys*, 11(2):121–137.

Compton, P. and Jansen, R. (1990). A philosophical basis for knowledge acquisition. *Knowledge Acquisition*, 2(3):241–258.

Crease, P. and Reichenbacher, T. (2011). Designing usable cartographic representations of geographic relevance for LBS users. In *Proc. 25th International Cartographic Conference*, number Tversky 2005, page 11, Paris.

de Berg, M., Bose, P., Cheong, O., and Morin, P. (2004). On simplifying dot maps. *Computational Geometry*, 27(1):43–62.

De Sabbata, S. and Reichenbacher, T. (2012). Criteria of geographic relevance: an experimental study. *International Journal of Geographical Information Science*, 26(8):1495–1520.

Delarze, R. and Gonseth, Y. (2008). *Lebensräume der Schweiz*. Ökologie - Gefährdung - Kennarten. Ott Verlag, Thun, Switzerland.

Devogele, T., Trevisan, J., and Raynal, L. (1996). Building a multi-scale database with scale-transition relationships. In *International Symposium on Spatial Data Handling*, pages 337–351. Citeseer.

Dorling, D. (1996). *Area cartograms: their use and creation*. Number 59 in CATMOGS (Concept and Techniques in Modern Geography). Quantitative Methods Study Group, Durham, UK.

Dougenik, J. a., Chrisman, N. R. N., and Niemeyer, D. R. (1985). An Algorithm To Construct Continuous Area Cartograms. *The Professional Geographer*, 37(1):75–81.

Douglas, D. H. and Peucker, T. K. (1973). Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 10(2):112–122.

Duckham, M., Kulik, L., and Birtley, A. (2006). A spatiotemporal model of strategies and counter strategies for location privacy protection. *Lecture Notes in Computer Science*, 4197:47–64.

Dutton, G. (1997). Digital map generalization using a hierarchical coordinate system. In ACSM-ASPRS, B. M., editor, *Auto Carto*, pages 367–376, Seattle, WA.

Dutton, G. (1999a). *A hierarchical coordinate system for geoprocessing and cartography*, volume 79 of *Lecture Notes in Earth Sciences*. Springer Berlin, Berlin, 1 edition.

Dutton, G. (1999b). Scale, Sinuosity, and Point Selection in Digital Line Generalization. *Cartography and Geographic Information Science*, 26(1):33–54.

Ebdon, D. (1985). *Statistics in geography*. Wiley-Blackwell, Oxford, UK, 2nd edition.

Edwardes, A. (2007). *Re-placing Location: Geographic Perspectives in Location Based Services*.

Phd, University of Zurich.
Edwardes, A. and Burghardt, D. (2003). Webpark location based services for species search in recreation area. In *Proceedings of the 21st International Cartographic Conference*, number 2, pages 1012–1021, Durban, South Africa.
Edwardes, A., Burghardt, D., and Weibel, R. (2005). Portrayal and Generalisation of Point Maps for Mobile Information Services. In Reichenbacher, T., Meng, L., and Zipf, A., editors, *Map-based mobile services: theories, methods and implementations*, chapter 2, pages 11–28. Springer, Berlin Heidelberg, 1 edition.
Ehrliholzer, R. (1995). Quality assessment in generalization: integrating quantitative and qualitative methods. In *Proceedings of the 17th International Cartographic ...*, pages 2241–2250, Barcelona, Spain. Institut Cartogràfic de Catalunya.
Eidgenossenschaft (2000). Verordnung über den Natur- und Heimatschutz (NHV) - Anhang 2. *NHV: Schutzstatus nach Eidgenossenschaft*, SR 451.1.
Ester, M., Kriegel, H., Sander, J., and Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In *Second International Conference on Knowledge Discovery and Data Mining*, pages 226–231, Portland, Oregon.
Estivill-Castro, V. (2002). Why so many clustering algorithms: a position paper. *ACM SIGKDD Explorations Newsletter*, 4(1):65–75.
Fäh, D., Giardini, D., Kästli, P., Deichmann, N., Gisler, M., Schwarz-Zanetti, G., Alvarez-Rubio, S., Sellami, S., Edwards, B., Allmann, B., Bethmann, F., Wössner, J., Gassner-Stamm, G., Fritsche, S., and Eberhard, D. (2001). ECOS-09 Earthquake Catalogue of Switzerland Release 2011 Report and Database.
Fairbairn, D. and Taylor, G. (1995). Developing a variable-scale map projection for urban areas. *Computers \& Geosciences*, 21(9):1053–1064.
Follin, J., Bouju, a., Bertrand, F., and Boursier, P. (2005). Multi-resolution extension for transmission of geodata in a mobile context. *Computers & Geosciences*, 31(2):179–188.
Furnas, G. W. (1982). The FISHEYE view - a new look at structured files. Technical Report 4, Bell Laboratories Technical Memorandum 81-11221-9, Murray Hill, New Jersey 07974.
Gaffuri, J. (2011). Improving Web Mapping with Generalization. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 46(2):83–91.
Gahegan, M. N. (1989). An efficient use of quadtrees in a geographical information system. *International journal of geographical information systems*, 3(3):201–214.
Gastner, M. T. and Newman, M. E. J. (2004). Diffusion-based method for producing density-equalizing maps. *Proceedings of the National Academy of Sciences*, 101(20):7499.
Glover, E. and Mackaness, W. (1999). Dynamic generalisation from single detailed database to support web based interaction. In *Proceedings 19th Int. Cartographic Conference*, volume 14, pages 1175–1184, Ottawa, Canada.
Goodchild, M. F. (2010). Twenty years of progress: GIScience in 2010. *Journal of Spatial Information Science*, 1(1):3–20.
Grünreich, D. (1992). ATKIS - A topographic information system as a basis for GIS and digital cartography in Germany. In Vinken, R., editor, *From Digital Map Series to Geo-Information Systems to Geo-Information Systems*, volume 122 of *Geologisches Jahrbuch Series A*, pages 207–216. Hannover, Germany.
Gusein-Zade, S. and Tikunov, V. (1993). A New Technique for Constructing Continuous Cartograms. *Cartography and Geographic Information Systems*, 20(3):66–85.
Guttman, A. (1984). R-trees: A dynamic index structure for spatial searching. In *Proceeding SIGMOD '84 Proceedings of the 1984 ACM SIGMOD international conference on Management*

of data, volume 13, pages 47–57. ACM New York, NY, USA.

Harrie, L., Sarjakoski, L. T., and Lehto, L. (2002a). A mapping function for variable-scale maps in small-display cartography. *Journal of Geospatial Engineering*, 4(2):111–124.

Harrie, L., Sarjakoski, L. T., and Lehto, L. (2002b). A variable-scale map for small-display cartography. *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, 34(4):237–242.

Harrie, L. and Sarjakoski, T. (2002). Simultaneous graphic generalization of vector data sets. *GeoInformatica*, 6(3):233–261.

Harrie, L. and Stigmar, H. (2010). An evaluation of measures for quantifying map information. *ISPRS Journal of Photogrammetry and Remote Sensing*, 65(3):266–274.

Harrie, L., Stigmar, H., Koivula, T., and Lehto, L. (2004). An algorithm for icon placement on a real-time map. In Fisher, P., editor, *Developments in Spatial Data Handling*, pages 493 – 507. Springer-Verlag.

Harrie, L. and Weibel, R. (2007). Modelling the overall process of generalisation. In *Challenges in the Portrayal of Geographic Information: Issues of Generalisation and Multi Scale Representation*, chapter 4, pages 67–87. Oxford, UK.

Haunert, J. and Sester, M. (2005). Propagating updates between linked datasets of different scales. In *Proceedings of XXII International Cartographic Conference*, number July, pages 11–16, Coruna, Spain.

Hoppe, H. (1996). Progressive meshes. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, volume pp, pages 99–108. ACM.

International Cartographic Association, I. (1973). *Multilingual Dictionary of Technical Terms in Cartography*. Franz Steiner Verlag, Wiesbaden.

Jakobsson, A. (2003). User requirements for mobile topographic maps. Technical report, National Land Survey of Finland, Helsinki, Finland.

Joao, E. (1995). The importance of quantifying the effects of generalization. In Müller, J.-C., Lagrange, J., and Weibel, R., editors, *GIS and Generalization. Methodology and Practice*, chapter 13. Taylor & Francis, London, UK.

Jones, C. (1997). *Geographical information systems and computer cartography*. Prentice Hall, London.

Jones, C. and Abraham, I. (1987). Line Generalisation In A Global Cartographic Database. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 24(3):32–45.

Jones, C. B. and Ware, M. J. (2005). Map generalization in the Web age. *International Journal of Geographical Information Science*, 19(8-9):859–870.

Kadmon, N. and Shlomi, E. (1978). A polyfocal projection for statistical surfaces. *The Cartographic Journal*, 15(1):36–41.

Kass, M., Witkin, A., and Terzopoulos, D. (1987). Snakes: Active contour models. *International journal of computer vision*, 1(4):321–331.

Keim, D. A., Hao, M. C., Dayal, U., Janetzko, H., and Bak, P. (2009). Generalized scatter plots. *Information Visualization*, 9(4):301–311.

Keim, D. a., North, S. C., and Panse, C. (2004). CartoDraw: a fast algorithm for generating contiguous cartograms. *IEEE transactions on visualization and computer graphics*, 10(1):95–110.

Kocmoud, C. J. and House, D. (1998a). A constraint-based approach to constructing continuous cartograms. In *8th International Symposium on Spatial Data Handling, SDH*, number 409, pages 236–246, Vancover.

Kocmoud, C. J. and House, D. H. (1998b). Continuous cartogram construction. *Proceedings Visu-*

alization '98 (Cat. No.98CB36276), pages 197–204.

Kresse, W. and Danko, D. M., editors (2012). *Springer Handbook of Geographic Information*. Springer Berlin Heidelberg, Berlin, Heidelberg.

Lam, H., Munzner, T., and Kincaid, R. (2007). Overview use in multiple visual information resolution interfaces. *IEEE transactions on visualization and computer graphics*, 13(6):1278–85.

Lang, T. (1969). Rules for robot draughtmen. *Geographical Magazin*, 62(1):50–51.

Lehto, L. and Kilpeläinen, T. (2001). Generalizing XML-encoded spatial data on the web. In *Proceedings of 20th International Cartographic Conference*, volume 4, pages 2390–2396.

Lehto, L. and Sarjakoski, L. T. (2005). Real-time generalization of XML encoded spatial data for the Web and mobile devices. *International Journal of Geographical Information Science*, 19(8-9):957–973.

Li, Z. and Huang, P. (2002). Quantitative measures for spatial information of maps. *International Journal of Geographical Information Science*, 16(7):699–709.

Lonergan, M. and Jones, C. (2001). An Iterative Displacement Method for Conflict Resolution in Map Generalization. *Algorithmica*, 30(2):287–301.

MacEachren, A. M. and Kraak, M.-J. K. (1997). Exploratory cartographic visualization: Advancing the agenda. *Computers & Geosciences*, 23(4):335–343.

Mackaness, W. and Ruas, A. (2007). Evaluation in the map generalisation process. In Mackaness, W. A., Ruas, A., and Sarjakoski, L. T., editors, *Generalisation of Geographic Information Cartographic Modelling and Applications*, chapter 5, pages 89–111. Elsevier Science Ltd, Oxford, UK.

Mackaness, W., Sarjakoski, L. T., and Ruas, A. (2007). *Generalisation of geographic information: cartographic modelling and applications*. Elsevier Science Ltd, Oxford, UK.

Mackaness, W. A. (2006). Automated Cartography in a Bush of Ghosts. *Cartography and Geographic Information Science*, 33(4):245–256.

Mackaness, W. A. and Purves, R. S. (2001). Automated Displacement for Large Numbers of Discrete Map Objects. *Algorithmica*, 30(2):302–311.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In LeCam, L. and Neyman, J., editors, *Proceedings of the fifth Berkeley symposium on ...*, volume 1, pages 281–297, Berkeley, California. University of California Press.

Mannes, J. (2004). *Dynamische kartographische Visualisierung von Location Based Services mittels SVG*. Msc thesis, University of Zurich.

Mark, D. (2003). Geographic information science: Defining the field. In Duckham, M., Goodchild, M., and Worboys, M., editors, *Foundations of geographic information science*, chapter 1, pages 1–15. Taylor & Francis, London, UK and New York.

Maso, J., Pomakis, K., and Julia, N. (2010). OpenGIS Web Map Tile Service Implementation Standard.

McMaster, R. (1987). Automated line generalization. *Cartographica*, 24(2):74.

McMaster, R. and Shea, K. (1992). *Generalization in digital cartography*. Association of American Geographers Washington, DC, Washington.

Meijers, M. (2011). *Variable-scale Geo-information*. Phd, University of Technology, Delft, The Netherlands.

Meijers, M. and van Oosterom, P. (2011). The space-scale cube: An integrated model for 2D polygonal areas and scale. In *28th Urban Data Management Symposium (UDMS 2011)*, volume XXXVIII, pages 95–102, Delft, The Netherlands.

Miller, C. C. (2006). A Beast in the Field: The Google Maps Mashup as GIS/2. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 41(3):187–199.

Monmonier, M. S. (1977). Nonlinear Reprojection to Reduce the Congestion on Symbols on Thematic Maps. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 14(1):35–47.

Mote, K. D. (2007). *Fast Point-Feature Label Placement for Dynamic Visualizations*. Master, Washington State University.

Mountjoy, D. N. (2001). *Perception-Based Development and Performance Testing of a Non-Linear Map Display*. Phd, North Carolina State University.

Müller, J., Lagrange, J., Weibel, R., and Salgé, F. (1995). Generalization: state of the art and issues. In *GIS and Generalization Methodoloygy and Practice*, chapter 1, pages 3–17. Taylor & Francis, London, 1 edition.

Mustafa, N., Krishnan, S., Varadhan, G., and Venkatasubramanian, S. (2006). Dynamic simplification and visualization of large maps. *International Journal of Geographical Information Science*, 20(3):273–302.

Neumann, J. (1994). The topological information content of a map/an attempt at a rehabilitation of information theory in cartography. *: The International Journal for Geographic Information*, 31(1):26–34.

Neun, M. (2007). *Data Enrichment for Adaptive Map Generalization Using Web Services*. Phd, University of Zurich, Zurich.

Neun, M., Burghardt, D., and Weibel, R. (2008a). Automated processing for map generalization using web services. *GeoInformatica*, 13(4):425–452.

Neun, M., Burghardt, D., and Weibel, R. (2008b). Web service approaches for providing enriched data structures to generalisation operators. *International Journal of Geographical Information Science*, 22(2):133–165.

Nivala, S., Sarjakoski, L. T., and Sarjakoski, T. (2007). Usability methods' familiarity among map application developers. *International Journal of Human-Computer Studies*, 65(9):784–795.

Nöllenburg, M., Merrick, D., Wolff, A., and Benkert, M. (2008). Morphing polylines: A step towards continuous generalization. *Computers, Environment and Urban Systems*, 32(4):248–260.

O'Sullivan, D. and Unwin, D. J. (2010). *Geographic Information Analysis*. John Wiley & Sons, Inc., Hoboken, NJ, USA.

Peters, S. (2013). Quadtree- and octree-based approach for point data selection in 2D or 3D. *Annals of GIS*, 19(1):37–44.

Petzold, I., Burghardt, D., and Bobzien, M. (2006). Workflow management and generalisation services. In *9th ICA Workshop on Generalization and Multiple Representation*, page 12, Portland.

Peuquet, D. J. (1984). A conceptual Framework and Comparison of Spatial Data Models. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 21(4):66–113.

Pietriga, E., Bau, O., and Appert, C. (2010). Representation-independent in-place magnification with sigma lenses. *IEEE transactions on visualization and computer graphics*, 16(3):455–67.

Pindat, C., Pietriga, E., Chapuis, O., and Puech, C. (2012). JellyLens: Content-Aware Adaptive Lenses. In *Proceedings of the 25th annual ACM symposium on User interface software and technology - UIST '12*, volume 2012, page 261, New York, New York, USA. ACM Press.

Plazanet, C. (1995). Measurement, characterization and classification for automated line feature generalization. In *Autoc-Carto 12*, pages 59–68, Charlotte, USA. Citeseer.

Plazanet, C. (1996). *Analyse de la géométrie des objets linéaires pour l'enrichissement des bases de données. Intégration dans le processus de généralisation cartographique des routes.* PhD thesis, Université Marne la Vallée.

Prim, R. (1957). Shortest connection networks and some generalizations. *Bell system technical journal*, 36(6):1389–1401.

Purvis, W. (2000). *Lichens*. Smithonians Natural World Series. Smithsonian Institution Press in Association with The Natural History Museum, Washington DC, USA.

Raper, J. (2007). Geographic relevance. *Journal of Documentation*, 63(6):836–852.

Rappo, A., Cecconi, A., and Burghardt, D. (2004). Fischaugenprojektionen für generalisierte Kartendarstellungen auf kleinen Bildschirmen. *Kartographische Nachrichten*, 2(2004):73–78.

Ratajski, L. (1967). Phénomènes des points de generalisation. In *International Yearbook of Cartography*, volume vii, pages 143–152. Güntersloh Bertelsmann, Bonn-Bad Godesberg.

Raubal, M. and Panov, I. (2008). A Formal Model for Mobile Map Adaptation. Lecture Notes in Geoinformation and Cartography, pages 11 – 34. Springer.

Regnauld, N. (2001). Contextual Building Typification in Automated Map Generalization. *Algorithmica*, 30(2):312–333.

Regnauld, N. and McMaster, R. (2007). A synoptic view of generalisation operators. In Mackaness, W. A., Ruas, A., and Sarjakoski, L. T., editors, *Generalisation of Geographic Information Cartographic Modelling and Applications*, chapter 3, pages 37–66. Elsevier Science Ltd, Oxford, UK.

Reichenbacher, T. (2004). *Mobile Cartography - Adaptive Visualisation of Geographic Information on Mobile Devices*. Phd thesis, Technischen Universität München.

Reichenbacher, T. and De Sabbata, S. (2011). Geographic relevance. *SIGSPATIAL Special*, 3(2):67–70.

Rhind, D. W. (1983). *A Census User's Handbook*. Methuen, London, UK.

Robbins, D. C., Cutrell, E., Sarin, R., and Horvitz, E. (2004). ZoneZoom. In *Proceedings of the working conference on Advanced visual interfaces - AVI '04*, page 231, Gallipoli, Italy. ACM Press.

Roth, R. (2011). A typology of operators for maintaining legible map designs at multiple scales. *Cartographic Perspectives*, 68:29–64.

Roth, R. E., Stryker, M., and Brewer, C. A. (2007). A typology of multi-scale mapping operators. In *Proceedings of GIScience 2008*, number 1996, pages 2007–2007, Park City, Utah. Pennstate Department of Geography.

Ruas, A. (1998). A method for building displacement in automated map generalization. *International Journal of Geographic Information Science*, 12(8):789–803.

Ruas, A. (2001). Automatic generalisation project: learning process from interactive generalisation. Technical Report 39, OEEPE Official Publication, Frankfurt am Main.

Ruas, A. and Plazanet, C. (1996). Strategies for automated generalization. In Kraak, M. and Mollenaar, M., editors, *Proceedings of 7th International Symposium on Spatial Data Handling*, Advances in GIS Research II, pages 319–336, Delft, The Netherlands. Taylor & Francis.

Rusak Mazur, E. and Castner, H. (1990). Horton's ordering scheme and the generalisation of river networks. *The Cartographic Journal*, 27(2):104–112.

Sabo, M. N., Bédard, Y., Moulin, B., and Bernier, E. (2008). Toward Self-Generalizing Objects and On-the-Fly Map Generalization. *Cartographica: The International Journal for Geographic Information and Geovisualization*, 43(3):155–173.

Samet, H. (1981). Neighbor Finding in Quadtrees. In *Proc. IEEE PRIP-81*, volume 18, pages 68–74, Dallas, TX, USA.

Samet, H. (1984). The Quadtree and Related Hierarchical Data Structures. *ACM Computing Surveys*, 16(2):187–260.

Samet, H. (1989). *Applications of Spatial Data Structures: Computer Graphics, Image Processing, and GIS*. Addison-Wesley Series in Computer Science. Addison-Wesley Longman Publishing Co., Inc., Reading, MA, USA.

Samet, H. (1990). *The Design and Analysis of Spatial Data Structures*. Addison-Wesley Series in Computer Science. Addison-Wesley Longman Publishing Co., Inc., Reading, MA, michael a. edition.

Samet, H. (2006). *Foundations of Multidimensional and Metric Data Structures*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

Sarjakoski, L. T. (2007). Conceptual models of generalisation and multiple representation. In Mackaness, W. A., Ruas, A., and Sarjakoski, L. T., editors, *Generalisation of geographic information: cartographic modelling and applications*, chapter 2, pages 11–35. Elsevier Science Ltd, Oxford, UK.

Sarjakoski, T. and Sarjakoski, L. T. (2005). The GiMoDig public final report. Technical report.

Sarkar, M. and Brown, M. H. (1994). Graphical fisheye views. *Communications of the ACM*, 37(12):73–83.

Schafer, W. and Bowman, D. (2003). A comparison of traditional and fisheye radar view techniques for spatial collaboration. *Proceedings of Graphics Interface*.

Scheidegger, C., Clerc, P., Dietrich, M., and Frei, M. (2002). *Rote Liste der gefährdeten baum- und erdbewohnenden Flechten der Schweiz*. BUWAL-Reihe Vollzug Umwelt. Bundesamt für Umwelt, Wald und Landschaft BUWAL; Eidgenössische Forschungsanstalt WSL; Conservatoire et Jardin botaniques de la Ville de Genève CJBG, Bern, Switzerland.

Schylberg, L. (1993). *Computational methods for generalization of cartographic data in a raster environment*. Phd, Royal Institute of Technology, Stockholm, Sweden.

Sester, M. (2000). Generalization based on least squares adjustment. *International Archives of Photogrammetry and Remote Sensing*, 33(B4/3; PART 4):931–938.

Sester, M. and Brenner, C. (2004). Continuous generalization for visualization on small mobile devices. In Altan, O., editor, *Proc. Conference on Spatial Data Handling*, volume 35, pages 469–480, Istanbul. Springer.

Sester, M. and Brenner, C. (2008). A vocabulary for a multiscale process description for fast transmission and continuous visualization of spatial data. *Computers & Geosciences*, 35(11):2177–2184.

Sester, M., Jokar Arsanjani, J., Klammer, R., Burghardt, D., and Haunert, J.-H. (2014). Integrating and generalising volunteered geographic information. In Burghardt, D., DuchÃªne, C., and Mackaness, W., editors, *Abstracting Geographic Information in a Data Rich World*, Lecture Notes in Geoinformation and Cartography, pages 119–155. Springer International Publishing.

Shneiderman, B. (1996). The eyes have it: a task by data type taxonomy for information visualizations. In *Proceedings 1996 IEEE Symposium on Visual Languages*, pages 336–343, Boulder, CO, USA. IEEE Comput. Soc. Press.

Silverman, B. (1986). Density estimation for statistics and data analysis. In *Monographs on Statistics and Applied Probability*, number 1951, page 176. Chapman & Hall/CRC, London.

Snyder, J. (1987). Magnifying-Glass Azimuthal Map Projections. *Cartography and Geographic Information Science*, 14(1):61–68.

Steiniger, S. (2007). *Enabling pattern-aware automated map Generalization*. Phd, University of Zurich.

Steiniger, S., Burghardt, D., and Weibel, R. (2006). Recognition of island structures for map generalization. In *Proceedings of the 14th annual ACM international symposium on Advances in geographic information systems*, page 74, New York, New York, USA. ACM.

Steiniger, S. and Weibel, R. (2007). Relations among Map Objects in Cartographic Generalization. *Cartography and Geographic Information Science*, 34(3):175–197.

Stigmar, H. and Harrie, L. (2011). Evaluation of Analytical Measures of Map Legibility. *The Cartographic Journal*, 48(1):41–53.

Stofer, S., Scheidegger, C., Clerc, P., Dietrich, M., Frei, M., Groner, U., Jakob, P., Keller, C., Roth, I., Vust, M., and Zimmermann, E. (2012). Nationales Daten- und Informationszentrum der Schweizer Flechten - SwissLichens. Birmensdorf, Schweiz. Eidgenössische Forschungsanstalt WSL, Datenbankauszug vom 25. September 2012 (SST_201209).

Stoter, J., Burghardt, D., Duchêne, C., Baella, B., Bakker, N., Blok, C., Pla, M., Regnauld, N., Touya, G., and Schmid, S. (2009). Methodology for evaluating automated map generalization in commercial software. *Computers, Environment and Urban Systems*, 33(5):311–324.

Sukhov, V. (1970). Application of information theory in generalization of map contents. *Application of information theory in generalization of map contents*, X:41–47.

Sun, S. (2013). An Optimized Rubber-Sheet Algorithm for Continuous Area Cartograms. *The Professional Geographer*, 65(1):16–30.

Taubin, G. (2001). Dual mesh resampling. In *Pacific Graphics*, volume 64, pages 94–113, Tokyo.

Tobler, W. (1986). Pseudo-cartograms. *The American Cartographer*, 13(1):43–50.

Tobler, W. R. (1973). A Continuous Transformation Userful for Districting. *Annals of the New York Academy of Sciences*, 219:215–220.

Töpfer, F. and Pillewizer, W. (1966). The principles of selection. *Cartographic Journal*, 3(1):10–16.

van Kreveld, M. (2001). Smooth generalization for continuous zooming. In *Proc. 20th International Cartographic Conference (ICC 2001)*, pages 2180–2185, Beijing, China.

van Oosterom, P. (1992). A storage structure for a multi-scale database: The reactive-tree. *Computers, Environment and Urban Systems*, 16(3):239–247.

van Oosterom, P. (1995). The GAP-tree , an approach to "on-the-fly" map generalization of an area partitioning. *GIS and Generalization: Methodology and Practise. Taylor & Francis, London*, (December):120–132.

van Oosterom, P. (2005). Variable-scale Topological Data Structures Suitable for Progressive Data Transfer: The GAP-face Tree and GAP-edge Forest. *Cartography and Geographic Information Science*, 32(4):331–346.

van Oosterom, P. and Meijers, M. (2011). Towards a true vario-scale structure supporting smooth-zoom. In *14th ICA/ISPRSWorkshop on Generalisation and Multiple Representation & the ISPRS Commission II/2 WG on Multiscale Representation of Spatial Data*, page 19, Paris.

van Oosterom, P. and Schenkelaars, V. (1995). The development of an interactive multi-scale GIS. *International Journal of Geographical Information Systems*, 9(4):489–489.

Venkateswaran, R. (2010). A Study of the Tourism Web Coverage in Switzerland. In Purves, R. and Weibel, R., editors, *Sixth international conference on Geographic Information Science - GIScience 2010*, number 2008, page 6, Zurich.

Venkateswaran, R. (2015). *Ad hoc Data Intergration for Mobile GIS Applications*. Phd, University of Zurich.

Venkateswaran, R. and Bereuter, P. (2009). User Adaptive Trip Planner. In *Ordnance Survey Geospatial Mashup Challenge - GISRUK 2009*, pages 1–5, Durham, UK.

Venkateswaran, R., Weibel, R., and Purves, R. S. (2014). Exploring and Visualizing Differences in Geographic and Linguistic Web Coverage. *Transactions in GIS*, pages n/a–n/a.

Viaña, R., Magillo, P., Puppo, E., and Ramos, P. a. (2006). Multi-VMap: A Multi-Scale Model for Vector Maps. *GeoInformatica*, 10(3):359–394.

Wang, W., Yang, J., and Muntz, R. (1997). STING : A Statistical Information Grid Approach to Spatial Data Mining. In *Proceedings 23th International Conference on Very Large Data Bases*, pages 186 – 195, Athens, Greece. Morgan Kaufmann Publishers, San Francisco, CA.

Ware, J. M., Jones, C. B., and Thomas, N. (2003). Automated map generalization with multiple operators: a simulated annealing approach. *International Journal of Geographical Information Science*, 17(8):743–769.

Wasserman, S. and Faust, K. (1994). *Social network analysis: Methods and applications*. Cambridge Univ Pr.

Waugh, T. (1986). A response to recent papers and articles on the use of quadtrees for geographic information systems. In *Proceedings of the Second International Symposium on Geographic Information Systems*, pages 33–37, Seattle, Washington, USA. International Geographical Union Commission on Geographical Data Sensing and Processing.

Weibel, R. (1991). Amplified Intelligence and Knowledge-Based Systems. In Buttenfield, B. P. and McMaster, R., editors, *Map generalization: making rules for knowledge representation*, pages 172–186. Longman, London, UK.

Weibel, R. (1995). Three essential building blocks for automated generalization. *GIS and Generalization: Methodology and Practice*, (1990):56–69.

Weibel, R. (1997). Generalization of spatial data: principles and selected algorithms. In van Kreveld, M., Nievergelt, J., Roos, T., and Widmayer, P., editors, *Algorithmic foundations of geographic information systems*, Lecture Notes in Computer Science, pages 99–152. Springer, Berlin, Germany.

Weibel, R. and Burghardt, D. (2008). Generalization, On-the-Fly. In Shekhar, S. and Xiong, H., editors, *Encyclopedia of GIS*, number 1991, pages 339–344. Springer, New York, 1 edition.

Weibel, R. and Dutton, G. (1999). Generalising spatial data and dealing with multiple representations. In Longley, P., Goodchild, M. F., Maguire, D. J., and Rhind, D. W., editors, *Geographical Information Systems: Principles, Techniques, Applications, and Management.*, chapter 10, pages 125–156. John Wiley, New York.

Wilson, I. D., Ware, J. M., and Ware, J. A. (2003). A Genetic Algorithm approach to cartographic map generalisation. *Computers in Industry*, 52(3):291–304.

Winter, S. and Tomko, M. (2004). Shifting the focus in mobile maps. *Proceedings of the International Joint Workshop*.

Wirth, V. (2010). Ökologische Zeigerwerte von Flechten - erweiterte und aktualisierte Fassung. *Herzogia*, 23(2):229–248.

Wood, J., Dykes, J., Slingsby, A., and Clarke, K. (2007). Interactive visual exploration of a large spatio-temporal dataset: reflections on a geovisualization mashup. *IEEE transactions on visualization and computer graphics*, 13(6):1176–83.

Yamamoto, D., Ozeki, S., and Takahashi, N. (2009). Focus+Glue+Context: an improved fisheye approach for web map services. In *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems - GIS '09*, pages 101–110, Seattle, Washington. ACM Press.

Yan, H. and Weibel, R. (2008). An algorithm for point cluster generalization based on the Voronoi diagram. *Computers & Geosciences*, 34(8):939–954.

Yang, B., Purves, R., and Weibel, R. (2007). Efficient transmission of vector data over the Internet. *International Journal of Geographical Information Science*, 21(2):215–237.

Zipf, A. and Richter, K.-F. (2002). Using focus maps to ease map reading. *Künstliche Intelligenz*, 4:35–37.